LIVRO DE EXERCÍCIOS EM BASIC

Jogos, testes e problemas para desenvolver a sua técnica

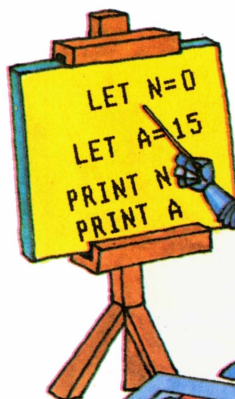
2.ª EDIÇÃO

EDITORA LUTÉCIA

LIVRO DE EXERCÍCIOS EM BASIC

Gaby Waters e Nick Cutler

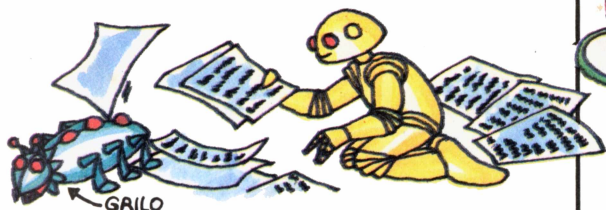
Tradução e Adaptação de
Ronaldo Sergio de Biasi

2.ª EDIÇÃO

Ilustrações de Naomi Reed
Diagramação de Graham Round
Compilação de Lisa Watts

LUTÉCIA

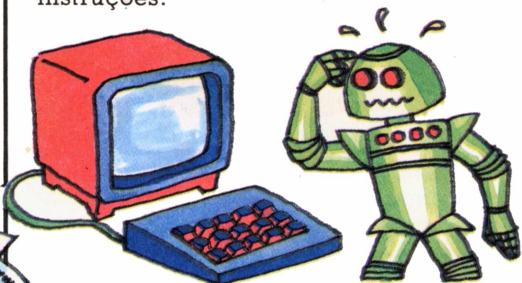
Sumário

Introdução

Este livro contém muitos exercícios, quebra-cabeças e problemas para ajudá-lo a praticar o seu BASIC. Existem programas com linhas e variáveis em branco para você completar, listagens cheias de erros para você localizar e corrigir e idéias de programas para você escrever sozinho. Este livro cobre as principais instruções de BASIC, começando com PRINT e terminando com uma orientação detalhada para você escrever um programa de caça ao tesouro.

GRILO

Os programas que contêm erros estão assinalados por um "grilo" como o que aparece na figura acima. Quando encontrar um "grilo", lembre-se de que terá que localizar e corrigir os erros para que o programa funcione. Em outros programas estão faltando linhas e variáveis. As linhas que faltam estão indicadas por um asterisco, e os espaços a serem completados, por pontos de interrogação.

No final do livro você encontrará uma discussão detalhada de todos os problemas e programas propostos. Se ficar atrapalhado, especialmente nos programas mais compridos, use as respostas como orientação. Muitas vezes, as primeiras linhas da resposta a um problema podem ajudá-lo a resolver outro problema ou a completar o programa sozinho.

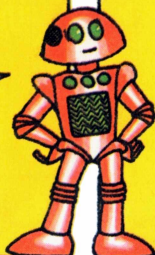
Pode ser que a listagem que aparece neste livro como resposta a um dos problemas seja diferente do programa que você escreveu. Se o seu programa funcionar corretamente, tudo bem; existem muitas maneiras de escrever o mesmo programa. Compare a sua resposta com a do livro para ver qual a que contém o menor número de instruções.

Todos os programas que aparecem neste livro foram escritos em um dialeto popular de BASIC e deverão funcionar, sem grandes alterações, na maioria dos microcomputadores. Algumas instruções variam de computador para computador; assim, se você tentar rodar um programa e receber uma mensagem de erro, verifique se as instruções de BASIC são as corretas para o modelo de micro que está usando. Para ajudá-lo, existem duas tabelas de conversão na página 47.

Se você tiver alguma dúvida a respeito das instruções de BASIC, consulte as páginas 45 e 46, onde são definidas todas as palavras de BASIC usadas neste livro. São dadas instruções especiais para escrever programas para micros da família Sinclair (TK-83, TK-85, CP-200) e também respostas adequadas para esses computadores.

Exercícios com PRINT

Nestas duas páginas, você vai praticar o uso da instrução PRINT, que diz ao micro para mostrar alguma coisa na tela. Você pode usar PRINT como um comando direto, isto é, sem número de linha. Nesse caso, o computador executará imediatamente a instrução. Depois de um comando direto, você tem que apertar RETURN (ou ENTER ou NEWLINE ou CR, dependendo do micro) para que sua ordem seja cumprida.

```
PRINT "PEIXE"
PRINT "2 PEIXES"
PRINT 2345
```

Não se esqueça de apertar RETURN depois de cada comando.

```
PRINT 2+2+3
PRINT 6*8
PRINT 15-4
PRINT 16/4
PRINT SQR(16)
PRINT 5346-257
```

Experimente escrever os comandos diretos acima no seu micro. Quando você diz ao computador para mostrar letras, ou letras e números misturados, eles devem ser colocados entre aspas na instrução. Números sozinhos não precisam de aspas.

Você também pode usar PRINT para fazer contas. A figura acima mostra algumas operações simples usando os símbolos matemáticos do BASIC.

```
PRINT "BOM DIA","BOM DIA"
PRINT "BOM DIA";"BOM DIA"
PRINT "BOM DIA ";" BOM DIA"
PRINT 555,777
```

```
PRINT "      BOM DIA"
PRINT SPC(10)"BOM DIA"
PRINT TAB(15);"BOM DIA"
PRINT TAB(10);555
```

Todas as instruções acima dizem ao micro em que lugar da tela deve começar a mensagem. Experimente para ver a diferença. Os sinais de pontuação têm significados especiais em BASIC. A vírgula diz ao computador para deixar alguns espaços em branco, e o ponto-e-vírgula para mostrar a palavra ou número seguinte na mesma linha, sem deixar nenhum espaço.

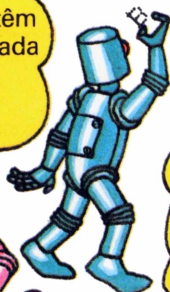
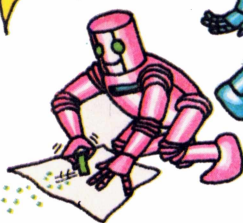
Como corrigir erros e alterar programas

Os erros nos programas são chamados de "grilos". Podem ser causados por erros na hora de copiar os programas. Você precisa saber como corrigir erros e alterar programas no seu micro.

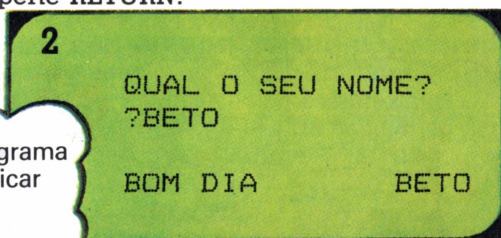
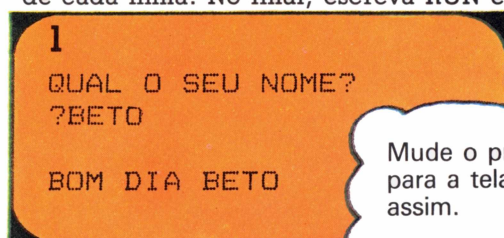
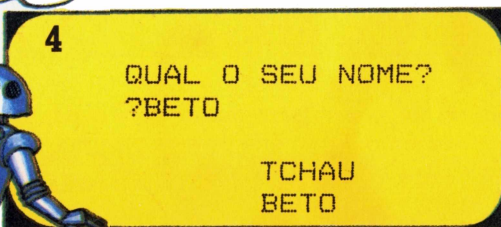
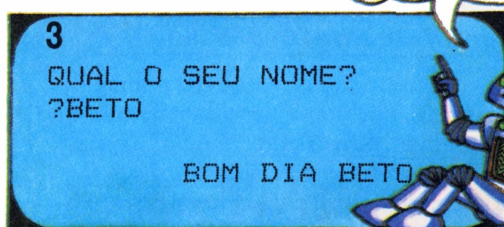
Muitos micros têm uma tecla chamada DELETE ou RUBOUT para corrigir erros.

OM DIA

Para acrescentar uma letra, você precisa mover o cursor. Consulte o manual do seu micro para saber como usar as teclas de controle do cursor.

Para apagar uma linha inteira, escreva apenas o número da linha e aperte RETURN.

Alguns quebra-cabeças

Eis um programa simples para você experimentar. Ele faz o computador exibir a mensagem que aparece na figura abaixo, à direita. Rode o programa e depois veja se consegue modificá-lo para que a mensagem fique como nas telas abaixo. Em caso de dúvida, consulte os comandos diretos da página anterior.

Escreva o programa no micro, apertando **RETURN** (ou a tecla equivalente) depois de cada linha. No final, escreva **RUN** e aperte **RETURN**.

A diferença entre o programa e os comandos diretos da página anterior é que cada linha de instruções em um programa tem um número. O computador guarda as instruções na memória e só as executa quando você escreve RUN. Se você numerar as linhas de dez em dez, poderá acrescentar novas instruções sem ter que numerar de novo todo o programa.

Você pode acrescentar novas linhas a um programa ou mudar linhas antigas, escrevendo-as de novo. Experimente escrever o programa acima e veja o que acontece quando você manda listá-lo e quando manda executá-lo.

Uso de variáveis

Variável é um espaço na memória do micro onde é armazenada uma informação. Para dizer ao computador para guardar uma informação em uma variável, você pode usar as instruções LET ou INPUT. Uma informação que consiste em palavras ou misturas de letras, números e símbolos é chamada de string. Uma string deve estar sempre entre aspas e o nome da variável correspondente deve terminar por um cifrão.

A instrução LET diz ao computador para rotular um espaço na memória e guardar alguma informação nesse espaço.

A instrução INPUT faz o computador esperar que você escreva a informação durante a execução do programa.

Como escolher os nomes das variáveis

Quais desses nomes são permitidos no seu micro?

```
LET B$="6422 RATOS"
LET B5=6422
LET K1=99
LET FL$="PULGA"
```

Alguns desses nomes de variáveis contêm palavras de BASIC e não são permitidos. Quais são eles?

```
LET LETRA$="BOM DIA!"
LET DEZ=10
LET RUN$="MAIS RATOS"
LET PULGA$="50 PULGAS"
```

Os nomes permitidos para as variáveis variam de micro para micro. O TK-83, por exemplo, só aceita nomes de uma letra para as variáveis string.*

Se você usar palavras como nomes de variáveis, não deve escolher nomes que contenham palavras de BASIC, para não confundir o computador. Experimente as linhas acima para ver quais as que o seu micro aceita.

Palavras e variáveis

Quando você coloca palavras e variáveis em uma instrução PRINT, as palavras devem estar entre aspas e deve haver um ponto-e-vírgula entre as palavras e a variável. Você também deve deixar um espaço depois de abrir ou antes de fechar as aspas.

Teste com PRINT

```
LET A=66
LET B=77
LET RR$="SAPOS-MARTELOS"
```

```
66 SAPOS-MARTELOS
COMERAM 77 MOSCAS
```

Escreva um programa usando as variáveis da tela da esquerda para que a tela do seu computador fique igual à tela da direita.

*Se o seu micro é um TK-83, você terá que mudar os nomes das variáveis string de mais de uma letra que aparecem neste livro.

Analisando um programa

O programa abaixo serve para converter temperaturas de graus Fahrenheit para Centígrados. Para entender melhor o que o programa faz, experimente acrescentar algumas instruções PRINT para fazer o computador mostrar os resultados intermediários. Assim fica mais fácil analisar qualquer programa.

```
10 INPUT A
20 LET F=32
30 LET B=5
40 LET C=B/9
50 LET D=A-F
60 LET R=D*C
70 PRINT R
```

A temperatura em graus Fahrenheit que você escreve fica armazenada na variável A.

Acrescente instruções para imprimir os valores de C e D.

Problema

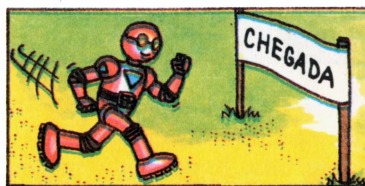
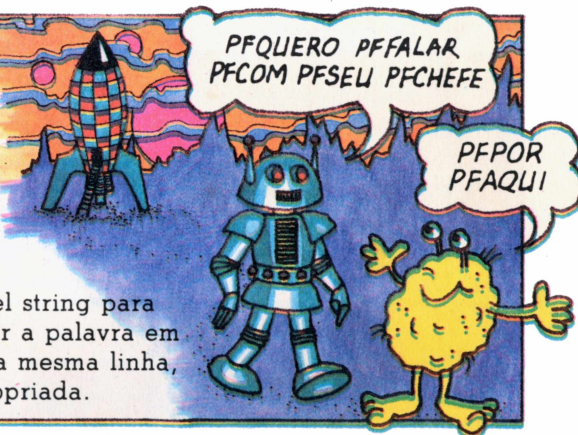
Mude o programa para que ele transforme graus Centígrados em Fahrenheit.

Idéias para programas

Abaixo você encontrará sugestões de alguns programas simples para escrever. Procure escolher nomes de variáveis que tenham alguma ligação com o que elas representam no programa. Escreva primeiro os programas no papel, depois introduza-os no micro e corrija os erros, se necessário.

1 Vamos falar pfiano?

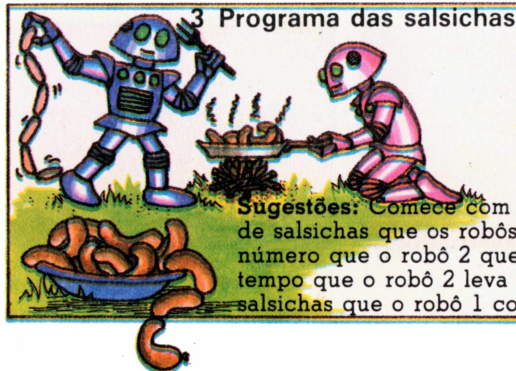
Você foi encarregado de negociar com os pfianos, que falam português, mas colocam as letras PF na frente de todas as palavras. Escreva um programa para traduzir palavras de português para pfiano.

□**Sugestões:** Use LET com uma variável string para guardar as letras PF e INPUT para pedir a palavra em português. Imprima as duas variáveis na mesma linha, depois de escrever uma mensagem apropriada.

2 Calculador de velocidades

Vamos fazer um programa para calcular velocidades? Você vai precisar de instruções INPUT para o tempo e a distância, de uma conta para calcular a velocidade e de instruções PRINT para imprimir uma mensagem apropriada e mostrar o resultado.

3 Programa das salsichas

O robô 1 come 30 salsichas por hora, mas o robô 2 só come 20 no mesmo tempo. Se o robô 2 quer comer no mínimo 35 salsichas e o robô 1 se recusa a comer mais devagar, quantas salsichas eles devem comprar e quanto tempo vão levar para comer todas?

Sugestões: Comece com três variáveis, duas para guardar os números de salsichas que os robôs podem comer por hora e uma para guardar o número que o robô 2 quer comer. Use novas variáveis para guardar o tempo que o robô 2 leva para comer 35 salsichas e o número de salsichas que o robô 1 come no mesmo tempo.

Repetições

É muito útil poder fazer o computador repetir alguma coisa várias vezes. Uma forma de conseguir isso é usando as instruções FOR...TO e NEXT. Experimente rodar o programa à direita para ver como funciona.

Você pode ver como a variável de um loop varia usando a instrução PRINT para observar os valores sucessivos da variável.

Testes com loops

Você sabe programar um loop para encher a tela de BOM DIA, e outro para escrever uma coluna de BOM DIA no centro da tela?

Uso de STEP

```
10 FOR I=1 TO 25 STEP 5
20 PRINT I
30 NEXT I
```

```
10 FOR J=20 TO 1 STEP-1
20 PRINT J
30 NEXT J
```

Você pode mudar a contagem de um loop usando a palavra STEP e um número que diz ao computador como a variável de contagem deve ser incrementada. Se você usar um número negativo, o computador contará para trás. Rode os programas acima e depois experimente mudar o número depois de STEP.

Escreva programas que façam a tela ficar como nas figuras acima.

Loops de espera

```
10 FOR J=10 TO 1 STEP-1
20 PRINT J
30 NEXT J
40 FOR K=1 TO 1000
50 NEXT K
60 PRINT "DECOLAR"
```

Um loop de espera é um loop sem instruções internas. No programa acima, as linhas 40 e 50 fazem o computador contar de 1 a 1000 antes de escrever a mensagem.

```
MENSAGEM SECRETA
MEMORIZE EM 5 SEGUNDOS
ANTES QUE DESAPARECA

ENCONTRE AGENTE X  NO AEROPORTO
```

Escreva um programa para fazer o seu micro escrever a mensagem acima e fazê-la desaparecer depois de 5 segundos. Você vai precisar de um loop de espera, seguido por uma instrução para o computador apagar a tela.

Desenhando na tela

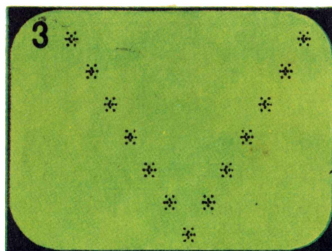
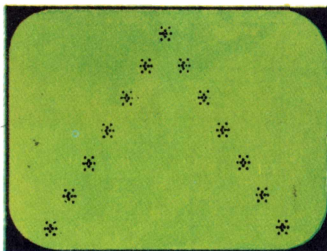
1

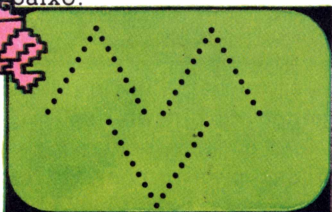
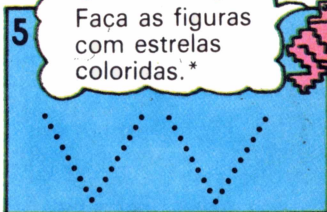
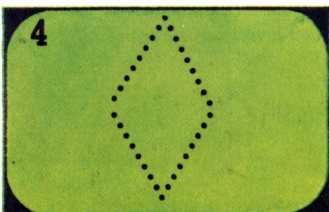
```
 10 LET A$="***"
 20 FOR J=1 TO 7
 30 PRINT TAB(J);A$
 40 NEXT J
 50 FOR K=1 TO 3
 60 PRINT TAB(J+1);A$
 70 NEXT K
 80 FOR L=7 TO 1 STEP-1
 90 PRINT TAB(L);A$
100 NEXT L
```

```
 80 FOR L=8 TO 18
 90 PRINT TAB(L);A$;"**"
100 NEXT L
```

O programa à esquerda faz um desenho com estrelas. Você pode mudar o desenho à vontade. A listagem acima é apenas uma sugestão.

2

```
10 CLS
20 LET A=15
30 PRINT TAB(A);"*"
40 FOR K=1 TO 9
*50 PRINT TAB(?);"*";
*60 PRINT TAB(?);"*"
*70 ?
```

Você é capaz de completar as linhas 50 a 70 deste programa para que a tela fique como na figura acima?

Agora mude o programa para mostrar a mesma forma de cabeça para baixo.

Você pode juntar os programas 2 e 3 para desenhar um losango.

Veja se consegue modificar os programas 2 e 3 para fazer figuras como estas.

*Consulte o manual do seu micro para saber como usar as instruções de cores.

Loops e mais loops

Você pode usar loops dentro de outros loops. Eles são chamados de loops internos. Cada vez que o loop de fora é repetido, o loop interno é executado um certo número de vezes.

```
10 FOR K=1 TO 3
20 FOR L=1 TO 5 ┐
30 PRINT K,L    ├── Loop interno
40 NEXT L       ┘
50 NEXT K
```

Este programa mostra os valores das variáveis dos loops, para que você veja como funciona o loop interno.

```
10 FOR I=1 TO 4
20 FOR J=1 TO 4
30 PRINT
40 NEXT J
50 PRINT "BOM DIA"
60 NEXT I
```

Rode este programa e depois experimente mudar o tamanho dos loops. Você é capaz de acrescentar outro loop interno para fazer o programa ficar mais lento?

Erros em loops

```
10 FOR L=1 TO 15
20 PRINT TAB(5);"*"·
   TAB(10);"*"
30 FOR J=1 TO 5
40 PRINT
50 NEXT J
60 NEXT L
```

```
10 FOR I=9 TO 0 STEP-1
20 FOR J=9 TO 0 STEP-1
30 FOR K=9 TO 0 STEP-1
40 PRINT I;J;K
50 NEXT I
60 NEXT J
70 NEXT K
```

Você deve ter cuidado para colocar todas as instruções do loop interno dentro do outro loop. O programa da esquerda está certo. Você sabe o que está errado no programa da direita? Como corrigi-lo?

Contador binário

```
10 FOR A=0 TO 1
20 FOR B=0 TO 1
30 FOR C=0 TO 1
40 FOR D=0 TO 1
50 PRINT D+C*2+B*4+A*8;" = ";
60 PRINT A;B;C;D
70 NEXT D
80 NEXT C
90 NEXT B
100 NEXT A
```

A linha 50 calcula o valor decimal de cada número binário.

O computador repete primeiro o loop mais interno.

Este programa usa quatro loops para contar em binário até 1111. Você é capaz de acrescentar mais loops e mudar as instruções PRINT para que ele possa contar até 11111111?

Mensagem que pisca

```
PERIGO
ATAQUE ESPACIAL
```

Você é capaz de escrever um programa para fazer piscar a mensagem acima? Vai precisar de um loop para apagar a tela e mostrar a mensagem e de dois loops de espera para a mensagem não piscar depressa demais.

Relógio digital

Faça um programa para que o micro funcione como um relógio. Você vai precisar de um loop para os segundos, um para os minutos e um loop de espera.

Decolagem de um foguete

Você é capaz de escrever um programa com vários loops para fazer "decolar" um foguete feito de estrelas?

1. Você precisa apagar a tela e imprimir muitas linhas em branco para que o foguete apareça na parte de baixo da tela.

```
  *            *
 ***          ***
 ***          ***
 ***          ***
** **       * *  * *
```

Dependendo do micro, você pode usar caracteres gráficos.

2. Você pode desenhar o foguete usando caracteres comuns em instruções PRINT.

3. Para fazer o foguete subir, acrescente mais linhas em branco e um loop de espera para que o foguete não ande depressa demais.

Ginástica

Você é capaz de escrever um programa para fazer uma figura como esta fazer ginástica, saltando na tela e movendo-se da esquerda para a direita?

Desenhe as figuras

Você pode desenhar uma figura simples em duas posições diferentes usando a instrução PRINT com letras e símbolos.

Esta linha em branco é para fazer o boneco pular.

```
PRINT
PRINT " O "
PRINT "<O>"
PRINT "/ \"
```

Escreva o programa

1. Você vai precisar de um loop do tamanho da largura da sua tela, que conte de 2 em 2. Você vai saber a razão quando tiver que escrever as instruções com PRINT e TAB.

2. Dentro do loop, você terá que apagar a tela antes de mostrar o primeiro boneco. Use TAB de tal forma que a posição da figura mude cada vez que for mostrada.

3. Use um loop de espera para a figura ficar algum tempo na tela.

4. Repita os passos acima com um TAB diferente para mostrar o segundo boneco.

```
<O>   O   <O>   O   <O>   O   <O>   O   <O>   O   <O>   O   <O>
 O  -<O>-  O  -<O>-  O  -<O>-  O  -<O>-  O  -<O>-  O  -<O>-  O
/ \-/ \-/ \-/ \-/ \-/ \-/ \-/ \-/ \-/ \-/ \-/ \-/ \
```

Exercícios com IF...THEN

A instrução IF...THEN é usada para comparar informações e dizer ao computador para fazer coisas diferentes de acordo com o resultado.

> = quer dizer maior que ou igual a.
< = quer dizer menor que ou igual a.

```
10 LET A=B
20 INPUT B
30 IF B=A THEN PRINT "IGUAIS"
40 IF B<>A THEN PRINT "DIFERENTES"
50 IF B<A THEN PRINT B;" E MENOR QUE ";A
60 IF B>A THEN PRINT B;" E MAIOR QUE ";A
```

Rode o programa acima para familiarizar-se com os símbolos que o computador usa para comparar informações.

```
10 PRINT "QUANTAS BATATAS FRITAS VOCE QUER?"
20 INPUT C
30 IF C>=0 AND C<=10 THEN
   PRINT "VAI MORRER DE FOME"
40 IF C>10 AND C<=25 THEN PRINT "OK"
50 IF C>25 AND C<=1000 THEN PRINT "ESGANADO"
60 IF C<0 OR C>1000 THEN PRINT "!!!"
```

Na maioria dos micros, você pode fazer várias comparações ao mesmo tempo, usando as palavras AND e OR. Rode o programa acima para ver o que acontece.

Tabuada do 13

```
10 LET A=13
20 FOR J=1 TO 13
30 PRINT "QUANTO E ";J;" X ";A;" ";
40 INPUT B
50 IF B=J*A THEN PRINT "CERTO"
60 NEXT J
```

Este é um programa para ver se você sabe a tabuada do 13. Você é capaz de acrescentar outras instruções IF...THEN para dizer que a resposta está errada e qual é a resposta certa?

Senha

```
10 LET S$="SALSICHAS"
20 PRINT "SENHA,POR FAVOR ";
30 INPUT P$
40 IF P$=S$ THEN PRINT "OK. PODE ENTRAR"
```

Você é capaz de completar este programa para que o computador imprima uma mensagem se você der a senha errada? Aumente o programa para o computador pedir também um número secreto.

Desvios

Você pode dar ao computador muitas instruções diferentes depois da palavra THEN. Assim, por exemplo, você pode dizer ao computador para parar ou pular para outra linha, usando a instrução GOTO.

```
10 LET C=0
20 PRINT "ESTA CANSADO?"
30 INPUT B$
40 LET C=C+1
50 IF B$="SIM" THEN STOP
60 IF C>10 THEN PRINT "DEVE ESTAR"
70 IF B$="NAO" THEN GOTO 20
80 PRINT "NAO DIGA BOBAGENS"
90 GOTO 20
```

Que acontece se você escrever "Bilubilu"?

Neste programa, o computador faz coisas diferentes, dependendo de sua resposta na linha 30. Rode o programa várias vezes, usando respostas diferentes.

Calculadora

```
DIGA UM NUMERO ? 7
DIGA OUTRO ? 11
VOCE QUER :
SOMAR, SUBTRAIR, DIVIDIR
OU MULTIPLICAR?
? MULTIPLICAR
A RESPOSTA E 77
```

A tela acima mostra a saída de um programa que pode somar, subtrair, multiplicar ou dividir dois números escolhidos. Você é capaz de escrever o programa?

Jogos de adivinhação

Acrescente algumas linhas para o micro dar uma pista da palavra.

1

```
10 INPUT X
20 CLS
30 PRINT "DIGA UM
   NUMERO"
40 INPUT Y
50 IF Y=X THEN GOTO
70
60 GOTO 30
70 PRINT "CERTO"
```

1. Faça o computador dizer se o número que você escolheu é grande demais ou pequeno demais.

2. Acrescente uma variável para contar quantos palpites você já deu. Estabeleça um limite para o número de palpites permitidos usando a instrução IF...THEN.

Este é um programa simples de adivinhação. Uma pessoa escolhe um número e outra tenta adivinhar qual é. Você é capaz de melhorar o programa, seguindo as sugestões acima?

2

```
PISTA
SABE RELINCHAR
? CAVALO
NAO
? EGUA
SIM
```

A tela acima mostra um jogo de adivinhação de palavras. Você é capaz de escrever o programa? É parecido com o de números, só que usa strings.

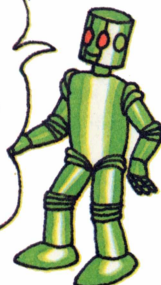
Corrida de cavalos

Aqui está a listagem de um jogo de corrida de cavalos... mas está incompleta. Veja se consegue colocar os números certos depois da instrução GOTO, nas linhas marcadas com um asterisco.

```
 10 LET N=0
 20 INPUT "PRIMEIRO:";H1
 30 INPUT "SEGUNDO:";H2
 40 CLS
 50 LET N=N+1
 60 INPUT "DIGA O PRIMEIRO:";G1
 70 INPUT "DIGA O SEGUNDO:";G2
* 80 IF G1=H1 AND G2=H2 THEN GOTO?
* 90 IF G1=H2 OR G2=H1 THEN GOTO?
*100 IF (G1=H1 AND G2<>H2) OR
     (G2=H2 AND G1<>H1) THEN GOTO?
110 PRINT "ERRADO"
*120 IF N=4 THEN GOTO?
130 PRINT "TENTE DE NOVO":GOTO 50
140 PRINT "UM PALPITE CERTO":GOTO 120
150 PRINT "CAVALO CERTO, LUGAR ERRADO":GOTO 120
160 PRINT "CERTO"
170 PRINT "PRIMEIRO:";H1;"-SEGUNDO:";H2
```


Como se joga

Existem seis cavalos numerados de 1 a 6. O primeiro jogador escolhe os cavalos que chegaram em primeiro e segundo lugares. O segundo jogador tem quatro tentativas para adivinhar quais são.

Idéias para melhorar o jogo

Números aleatórios

A instrução RND serve para gerar números aleatórios, mas a forma como é usada varia de micro para micro. Verifique no manual do seu micro ou na tabela de conversão do final do livro qual é a forma que deve utilizar. Depois, tente resolver os exercícios que aparecem nestas duas páginas.

```
PRINT RND(99)
PRINT RND(10)
```

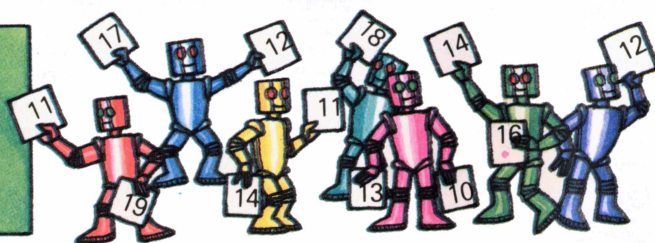
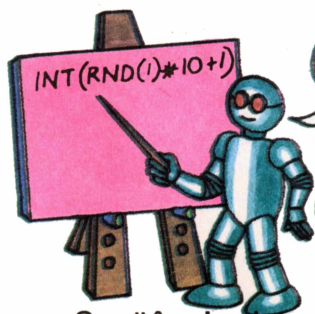
Em alguns micros, usa-se RND com um número entre parênteses para gerar um número inteiro entre 1 e o número entre parênteses.

```
PRINT INT(RND(1)*99+1)
PRINT INT(RND(0)*99+1)
PRINT INT(RND*99+1)
```

Verifique se o seu micro usa (1) ou (0) depois de RND.

Em muitos micros, para gerar um número inteiro você precisa usar a palavra INT com RND seguido de (1) ou (0), se necessário. Você deve multiplicar RND pelo número de números que deseja gerar e somar o menor número da série. Assim, por exemplo, todas as instruções acima são para o computador escolher um número ao acaso entre 1 e 99.

```
PRINT RND(6)+5
PRINT INT(RND(1)*6+5)
PRINT INT(RND(0)*6+5)
PRINT INT(RND*6+5)
```

As instruções acima são para gerar um número ao acaso entre 5 e 10. Uma delas deve servir para o seu micro.

Você é capaz de fazer o seu micro gerar números aleatórios entre 10 e 20?

Neste livro as instruções RND são escritas assim. Não se esqueça de modificá-las, caso seja necessário.

Quebra-cabeças

Examine o jogo de adivinhação de números da página 13 e veja se é capaz de modificá-lo para que o computador escolha um número entre 1 e 20 para você adivinhar.

Seqüência de números

```
DESCUBRA O NUMERO
SEGUINTE NESTA
SEQUENCIA
 4  13  22
? 31
CERTO
```

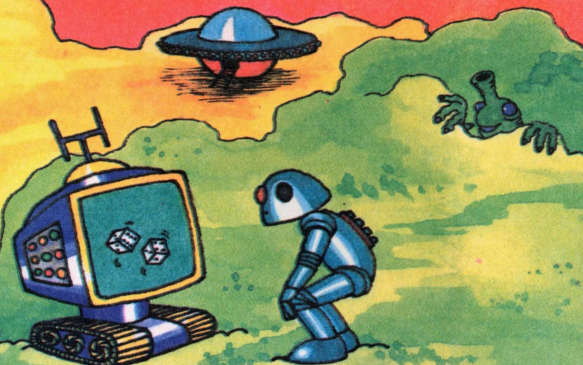
Veja se é capaz de escrever o programa para o jogo acima. Parte do programa aparece na listagem do meio.

```
*10 LET X=?
*20 LET Y=?
 30 FOR I=1 TO 3
 40 PRINT X+I*Y
 50 NEXT I
```

```
PRINT X+I*I
PRINT I*I-Y
PRINT X+Y-I*I
PRINT Y+X^I
```

Você vai ter que colocar os números corretos nas linhas 10 e 20 e acrescentar algumas instruções PRINT e IF...THEN. Na listagem da direita aparecem algumas idéias para mudar a linha 40 e produzir outras seqüências de números. Você sabe como mudar o programa para que o computador escolha uma seqüência ao acaso cada vez que você rodar o programa?

Fuga do planeta Zorgo

Veja se é capaz de escrever um programa para este jogo. Você está no planeta Zorgo e precisa de 50 créditos para comprar combustível suficiente para voltar à Terra. Você já tem 10 créditos, mas a única maneira de conseguir os outros 40 é jogar um jogo com um computador alienígena. As telas abaixo mostram como é o jogo.

```
VOCE TEM 10 CREDITOS
QUANTO VOCE APOSTA ?5
APERTE J PARA JOGAR ?J
      5   2
FICOU NA MESMA
```

```
VOCE TEM 10 CREDITOS
QUANTO VOCE APOSTA ?9
APERTE J PARA JOGAR ?J
      6   4
GANHOU O TRIPLO
VOCE TEM 28 CREDITOS
```

DADOS	SUA APOSTA
DOBRADINHA:	DOBRA
10 OU 11:	TRIPLICA
6 OU 7:	FICA IGUAL
OUTROS:	PERDE

Você aposta quantos créditos quiser. O computador joga os dados e você ganha ou perde uma certa quantia, de acordo com a tabela acima à direita.

Jogo do Papel, Pedra e Tesoura: Descubra os erros

Este é um programa para você jogar o jogo do Papel, Pedra e Tesoura com o computador, mas está cheio de erros. Com o auxílio dos comentários à direita da listagem, veja se consegue localizar os erros e corrigi-los.

```
 10 CLS
 20 LET C=0
 30 LET A=0
 40 LET F=0
 50 LET R=INT(RND(1)*4+1)
 60 IF R=1 THEN LET C$="PAPEL"
 70 IF R=2 THEN LET C$="PEDRA"
 80 IF R=3 C$="TESOURA"
 90 PRINT "ESTOU PRONTO"
100 PRINT "VOCE QUER PAPEL, PEDRA OU TESOURA"
110 INPUT A$
120 PRINT
130 IF C$="PAPEL" AND A$="TESOURA" THEN LET F=1
140 IF C$="PEDRA" AND A$="PAPEL" THEN LET F=1
150 IF A$="TESOURA" AND A$="PEDRA" THEN LET F=1
160 IF C$=A$ THEN LET F=2
170 PRINT "VOCE ESCOLHEU ";A$
180 PRINT "EU ESCOLHI ";C$
190 PRINT "PORTANTO
200 IF F=0 THEN PRINT "EU GANHEI"
210 IF F=F THEN PRINT "VOCE GANHOU"
220 IF F=2 THEN PRINT "EMPATOU"
230 IF F=0 THEN LET A=A+1
240 IF F=1 THEN LET C=C+1
250 PRINT "CONTAGEM :"
260 PRINT "EU :";C
270 PRINT "VOCE :";A
280 IF C>10 AND A>10 THEN GOTO 40
290 PRINT "ACABOU"
```

(Linhas 20–40) C é para a contagem do computador. A é para a contagem do jogador. F diz ao computador quem ganhou nas linhas 130-160.

(Linhas 60–80) A escolha do computador, que depende do valor de R, é guardada em C$.

(Linhas 130–160) O computador verifica quem ganhou. Se foi o jogador, faz F igual a 1. Se foi empate, faz F igual a 2. Se foi o computador, deixa F com o valor 0.

(Linhas 200–240) O computador diz quem ganhou e atualiza a contagem com base no valor de F.

Lembra-se de como é o jogo? O papel cobre a pedra, a pedra cega a tesoura e a tesoura corta o papel.

Mexendo com caracteres

O computador pode fazer muitas coisas com caracteres guardados em strings. Os programas desta página mostram algumas das instruções de BASIC para manipular strings. Se o seu micro é da família Sinclair, essas instruções são um pouco diferentes.

```
LET R$="COMPUTADOR"
PRINT LEFT$(R$,3)
PRINT RIGHT$(R$,3)

LET C$="SAPO-MARTELO"
PRINT LEFT$(C$,4)
PRINT RIGHT$(C$,7)
```

LEFT$** e **RIGHT$ dizem ao computador para separar um certo número de caracteres da string, começando pela esquerda ou pela direita, respectivamente. O número indica quantos caracteres devem ser tomados.

```
10 LET K$="CANGURU"
20 PRINT "ONDE FICA A"
30 PRINT "PRIMEIRA LETRA";
40 INPUT S
50 PRINT "QUANTAS LETRAS";
60 INPUT N
70 PRINT MID$(K$,S,N)
```

MID$ diz ao computador para separar letras do meio de uma string. O primeiro número dentro dos parênteses diz onde fica a primeira letra a ser tomada e o segundo o número de letras a tomar.

O computador conta os espaços como se fossem letras ou símbolos.

```
10 PRINT "DIGA UMA PALAVRA"
20 INPUT W$
30 LET L=LEN(W$)
40 PRINT "EXISTEM ";
50 PRINT L;" LETRAS";
60 PRINT " NA PALAVRA ";W$
```

A instrução **LEN** é usada para contar o número de caracteres de uma string. Experimente rodar o programa acima.

Micros da família Sinclair

```
LET A$="ABCDEFGHIJKLMNOP"
PRINT A$(4 TO 6)
DEF
PRINT A$(14 TO 16)
NOP
```

Os computadores da família Sinclair não usam as instruções **LEFT$**, **RIGHT$** e **MID$**. A maneira de manipular strings nesses micros está ilustrada acima.

Testes com strings

Você é capaz de substituir os pontos de interrogação da linha 40 para que sua tela fique assim? Pista: você pode usar **MID$** para separar um caractere de uma string de cada vez.

Faça o computador escrever uma palavra ao contrário usando **MID$** e um loop com incremento -1.

Usando LEFT$ e RIGHT$, veja se você consegue escrever a linha 50 do programa para que a tela fique assim.

A maior palavra

Na linha 50, você tem que dizer ao computador para guardar a maior palavra até o momento na variável A$. Experimente usar IF...THEN e LEN.

```
10 LET A$=""
20 PRINT "PALAVRAS"
30 FOR J=1 TO 5
40 INPUT W$
*50 ?
60 NEXT J
70 PRINT "MAIOR PALAVRA :"
80 PRINT A$
```

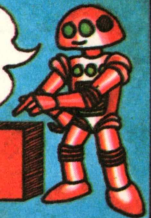
```
PALAVRAS
? GATO
? BOI
? HIPOPOTAMO
? COBRA
? FORMIGA
MAIOR PALAVRA :
HIPOPOTAMO
```

Este programa descobre qual é a maior palavra em uma lista de cinco. Complete a linha que falta.

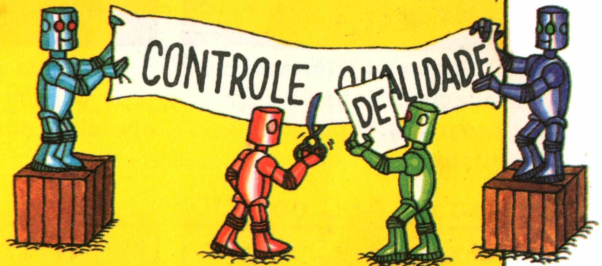
A menor palavra

Você é capaz de escrever um programa para descobrir a menor palavra? É muito parecido com o programa acima, só que na linha 10 você vai ter que colocar em A$ uma string mais comprida do que qualquer palavra a ser examinada. Também vai ter que mudar a instrução IF...THEN.

Você pode colocar qualquer coisa na variável A$.

```
LET A$="XXX!!!&&ABC*
**123!!!XXXXXXXXXX"
```

Editor de texto

A listagem abaixo mostra um programa que lhe permite escrever uma frase e depois mudar as palavras do texto. Antes de rodar o programa, você precisa completar as linhas assinaladas com um asterisco. As observações à direita do programa servirão para ajudá-lo.

```
 10 CLS
 20 PRINT "DIGA A FRASE"
 30 INPUT S$
 40 LET S$=?
 50 PRINT "PALAVRA PARA MUDAR";
 60 INPUT W$
 70 LET W$=?
 80 PRINT "PALAVRA NOVA";
 90 INPUT N$
*100 LET LS=?
*110 LET LW=?
 120 LET A$=""
 130 LET K=1
*140 IF MID$(S$,?,?)=W$ THEN
     LET A$=S$
*150 IF A$=S$ THEN LET S$=LEFT$
     (?,?)+?+RIGHT$(A$,LS-(K+LW-2))
 160 LET LS=LEN(S$)
 170 LET K=K+1
*180 IF K<=LS-LW+1 THEN GOTO ?
 190 PRINT S$
 200 GOTO 50
```

Para ver como o programa funciona, escreva uma frase em uma folha de papel e aplique as instruções do programa a essa frase.

Linhas 40 e 70: Faz o computador acrescentar um espaço no início e no final de S$ e W$ Você sabe para quê?

Linhas 100 e 110: Faz LS igual ao comprimento da frase (S$) e LW igual ao comprimento da palavra (W$).

Linha 140: Complete esta linha para fazer o computador procurar na frase (S$) a palavra (W$) a ser substituída. Pista: use K para contar os caracteres.

Linha 150: Esta linha faz o computador tomar os caracteres que estão à esquerda da palavra a ser substituída, acrescentar a nova palavra e completar com o resto da frase. Você é capaz de colocar as variáveis que estão faltando?

Outras instruções

Dentro do computador, os caracteres são representados por números de código. Você também pode usar esses números. A instrução CHR$ transforma um número em um caractere. A instrução ASC (ou CODE, nos micros da família Sinclair) faz o oposto, transformando um caractere no seu número de código. Todos os micros, com exceção do TK-83, usam um código-padrão, chamado ASCII*.

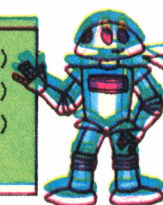
Uso de CHR$

Experimente algumas instruções CHR$ usando esses e outros números. A tabela completa dos números de código está no manual de seu micro.

Testes com letras

Coloque os números que faltam na linha 10 para o programa imprimir o alfabeto.

Escreva um programa para imprimir o alfabeto em letras minúsculas.

Escreva um programa para mostrar na tela letras escolhidas ao acaso.

Uso de ASC ou CODE

Experimente algumas instruções ASC (ou CODE) usando esses e outros caracteres para ver quais são os números de código.

Como comparar letras

Usando os sinais > e < tente escrever um programa para comparar duas letras e colocá-las em ordem alfabética.

Caixa alta/caixa baixa

```
 10 PRINT "ESCREVA SUA MENSAGEM"
 20 INPUT M$
 30 FOR J=1 TO LEN(M$)
 40 LET X$=MID$(M$,J,1)
*50 IF X$>="a" AND X$<="z" THEN
    PRINT CHR$(ASC(X$)-?);
*60 IF X$>="A" AND X$<="Z" THEN
    PRINT CHR$(ASC(X$)+?);
 70 IF X$<"A" OR X$>"Z" THEN PRINT X$;
 80 NEXT J
```

Complete os números que estão faltando nas linhas 50 e 60 para fazer o computador transformar letras maiúsculas em minúsculas e vice-versa. Você não conseguirá rodar este programa se o seu micro tiver apenas letras maiúsculas.

*ASCII é a abreviação de American Standard Code for Information Interchange (Código Americano Padrão para Intercâmbio de Informações).

Códigos secretos

Aqui estão as idéias para alguns programas para gerar mensagens em código. A figura à direita mostra como o primeiro programa funciona. Você só tem que completá-lo.

Código alfabético

O programa toma alternadamente a letra anterior e a letra seguinte do alfabeto.

Você sabe completar a última palavra?

```
-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
O AGENTE PARTE ESTA NOITE
N BFFMUD QZSSF DTSB M
```

Programa do código alfabético

```
  10 PRINT "DIGA SUA MENSAGEM"
  20 INPUT M$
* 30 FOR J=1 TO?
* 40 LET X=?
* 50 IF X<? OR X>? THEN LET N=X:
     GOTO 100
* 60 IF INT(J/2)=J/2 THEN LET N=X?1
* 70 IF INT(J/2)<>J/2 THEN LET N=X?1
* 80 IF N<? THEN LET N=N+26
* 90 IF N>? THEN LET N=N-26
*100 PRINT ?
 110 NEXT J
```

O loop é repetido tantas vezes quantos são os caracteres em M$. (linha 30)

Faz o computador tomar um caractere de cada vez e guardar o seu número de código ASCII em X*. (linha 40)

Verifica se o caractere é uma letra. (Neste código, os números e espaços não são alterados.) (linha 50)

Soma 1 a X se o contador do loop (J) é par e subtrai 1 se é ímpar. (linhas 60–70)

Se o novo número (N) está fora do alfabeto, resolve o problema somando ou subtraindo 26. (linhas 80–90)

Imprime a nova letra usando CHR$. (linha 100)

Código com número-chave

←N = X+Número-chave→N = N - 26

Alfabeto	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Alfabeto em código	GHIJKLMNOPQRSTUVWXYZABCDEF

Neste código, o alfabeto é deslocado de um certo número (N) de letras. No exemplo acima, N = 6. Você é capaz de escrever um programa que faça isso? Use o número-chave que quiser.

Código variável

```
J = 1 2 3 4 5 6 7 8 9 10 11 12
      O   A G E N T E     P A R
```

Neste código, o valor da variável do loop, J, é somado ao código ASCII de cada letra. Escreva dois programas, um para codificar a mensagem, outro para decodificá-la.

Código de inversão

O AGE NTE PARTE ESTA NOITE

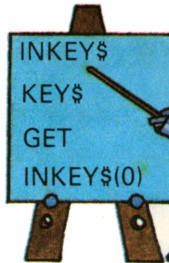
OGAN EETP RAETE TS AONTI E

Neste código, você divide a mensagem em pares de letras e troca as posições das letras do par, incluindo os espaços. Para escrever o programa, use um loop com incremento 2.

*Se o seu micro é um TK-83, use os números de código apropriados.

SEATE U AMM NEASEG MESRCTE AMEC DOGI OEDI VNREAS O

Exercícios com INKEY$

A instrução INKEY$ faz o computador consultar o teclado para ver se alguma tecla está sendo apertada. Ao contrário do que acontece com INPUT, o computador não espera por você, mas continua a executar o programa.

Estas são algumas das instruções usadas para consultar o teclado em diferentes computadores. Consulte o manual do seu micro ou a tabela da página 47.

```
10 LET A$=INKEY$
20 IF A$="" THEN PRINT "!";
30 IF A$<>"" THEN PRINT A$;
40 GOTO 10
```

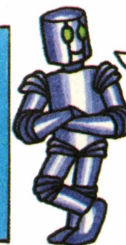
Experimente rodar este programa usando a instrução correta para o seu micro. Quando você aperta uma tecla, o computador imprime o caractere correspondente, caso contrário imprime um ponto de exclamação.

```
BOM DIA        BOM DIA       BOM D
IA        BOM DIA              BOM
 DIA BOM DIA BOM DIA             B
OM DIA        BOM DIA   BOM DIA
```

Escreva um programa que faça o computador imprimir BOM DIA quando você apertar uma tecla e um espaço quando não apertar, como na tela acima.

Fazendo o computador esperar

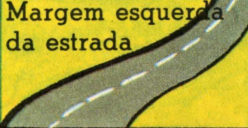
```
10 LET A$=INKEY$(50)
20 IF A$="" THEN
   PRINT " ";
30 IF A$<>"" THEN
   PRINT A$;
40 GOTO 10
```

```
10 LET N=0
20 LET A$=INKEY$
30 IF A$<>"" THEN GOTO 70
40 LET N=N+1
50 IF N<50 THEN GOTO 20
60 PRINT " ";:GOTO 10
70 PRINT A$;:GOTO 10
```

Às vezes é útil fazer o computador esperar um pouco antes de continuar o programa. Em algumas versões de BASIC, você tem que colocar um número entre parênteses depois de INKEY$, como na listagem da esquerda. Isto diz ao micro quanto tempo deve esperar (em frações de segundo) antes de continuar. Se o seu micro não permite isso, coloque o INKEY$ no interior de um loop, como na listagem da direita.

Descubra os erros

Será que você é capaz de descobrir os erros do programa abaixo e corrigi-los? O computador deve escolher dois números ao acaso entre 1 e 25. Você tem que somá-los mentalmente e apertar qualquer tecla assim que a resposta correta aparecer na tela.

```
10 CLS
20 PRINT "APERTE QUALQUER TECLA
30 PRINT "QUANDO A RESPOSTA CERTA
APARECER"
40 LET N=0
50 LET X=INT(RND(1)+25+1)
60 LET Y=INT(RND(1)*25+1)
70 PRINT
80 PRINT "X;" + ";Y;" = "
90 LET N=N+1
100 PRINT N
110 LET INKEY$=A$
120 IF A$<>"" THEN GOTO 180
130 FOR K=1 TO 100:NEXT K
140 IF N<30 THEN GOTO 90
150 PRINT "ERROU. A RESPOSTA E ";X+Y
160 FOR K=1 TO 1000:NEXT T
170 GOTO 30
180 IF N<>X+Y THEN GOTO 150
190 PRINT "ACERTOU. A RESPOSTA E X+Y"
```

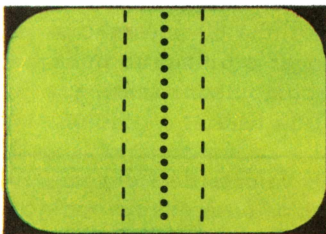
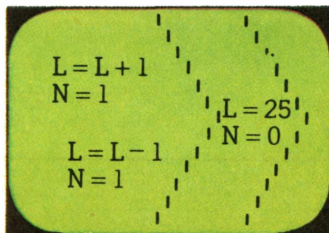
Este programa tem oito erros.

Como programar um exame de motorista

Seguindo os passos abaixo, veja se é capaz de programar este jogo. Para mostrar o carro e a estrada, use PRINT TAB com um * para o carro e ! para as margens da estrada. A estrada desce a tela em ziguezague enquanto você dirige o carro com duas teclas para mantê-lo na estrada.

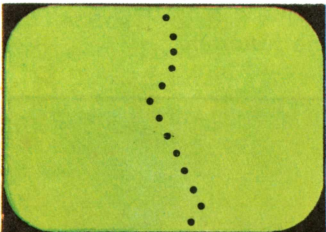
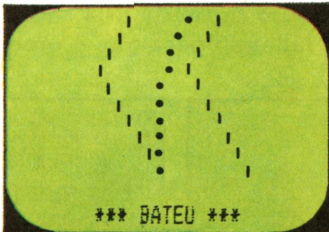
1. Definindo as variáveis

Você precisa de quatro variáveis, C,L,W e R, para posicionar o carro e as margens da estrada. Os valores iniciais vão depender do tamanho de tela do seu micro.

2. Desenhando a estrada

Você deve usar a linha LET R = L + W dentro do loop para fazer com que o valor de R acompanhe o valor de L.

Agora faça o computador mostrar a estrada e o carro, usando PRINT TAB com uma variável e um símbolo. Use GOTO para repetir as instruções e desenhar uma estrada reta.

Para fazer a estrada ziguezaguear, você tem que mudar o valor de L cada vez que a instrução PRINT TAB for executada. Para que a estrada não saia da tela, defina uma nova variável, N. Faça N = 1 se L = 1 e N = 0 se L = à largura da sua tela. Diga ao computador para somar ou subtrair 1 de L, dependendo do valor de N.

3. Dirigindo o carro

Você sabe programar um ziguezague aleatório?

Para dirigir o carro, você precisa de uma instrução INKEY$. Escolha duas teclas (como < e >) para somar ou subtrair 1 de C quando forem apertadas. Se o programa estiver muito rápido, use um loop de espera.

Finalmente, verifique se o carro bateu, comparando C com L e R. Nesse caso, avise ao motorista.

Quebra-cabeças com DATA

Uma das maneiras mais simples de fornecer ao computador grandes quantidades de informações é usando as instruções READ e DATA. Uma linha de DATA contém uma lista de palavras ou números e READ diz ao computador para guardar os dados em uma ou mais variáveis. O micro TK-83 não reconhece essas instruções.

```
10 FOR I=1 TO 6
20 READ X,X$
30 PRINT X;" ";X$
40 NEXT I
50 DATA 13,PEIXES,77,SAPOS,91,COBRAS
60 DATA 23,GATOS,62,BOIS,2,RATOS
```

Rode o programa para ver como funcionam as instruções READ e DATA. Os dados são guardados nas variáveis X e X$.

```
10 PRINT "ESCREVA SEU NOME"
20 INPUT N$
30 READ X$
40 IF X$=N$ THEN GOTO 70
*50 IF X$="?" THEN PRINT "SEU NOME NAO
ESTA NA LISTA":STOP
60 GOTO 30
70 PRINT "OK...SEU NOME ESTA
NA LISTA"
*80 DATA ?
```

Neste programa, o computador pergunta o seu nome e o compara com uma lista de nomes. Coloque quantos nomes quiser na linha 80, mas não se esqueça de colocar o último na linha 50, para que o computador saiba que chegou ao final da lista.

Uso de RESTORE

```
*10 FOR J= ? TO ?
*20 FOR I= ? TO ?
 30 READ N$
*40 IF LEFT$(N$,1)=CHR$( ? ) THEN PRINT N$
 50 NEXT I
 60 RESTORE
 70 NEXT J
 80 DATA VERA,XAVIER,ZACARIAS,HORACIO,GUILHERME,ALFREDO
 90 DATA TANIA,CAROLINA,LEANDRO,VANIA,ANDRE,DALILA
100 DATA RONALDO,MARILIA,SERGIO,CLAUDIO,SONIA
```

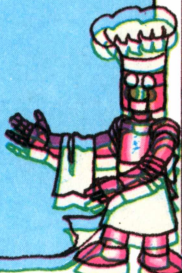
No programa acima, que serve para colocar uma série de nomes em ordem alfabética, os nomes estão em instruções DATA e a instrução RESTORE diz ao computador para voltar ao primeiro nome da lista cada vez que o loop de J é repetido. Complete o que está faltando para que o programa funcione.

Descubra os erros

1

```
10 FOR K=1 TO 6
20 READ N : PRINT N
30 NEXT K
40 DATA 01-232,22-36-41,341-2241/2
50 DATA 97-24-11,47-29-01,236-4013
```

Neste programa, vários números de telefone aparecem como dados. Você é capaz de descobrir o que está errado e corrigir o erro?

2

```
10 READ X
20 PRINT X
30 GOTO 10
40 DATA 1,461,892,66,1471,4462,1,3
50 DATA 53,80,241,90,371,825,33,13
```

Se não encontrar o erro deste programa, experimente rodá-lo no seu micro. O computador vai imprimir uma mensagem de erro. Qual a maneira mais simples de resolver o problema?

Bar Bado

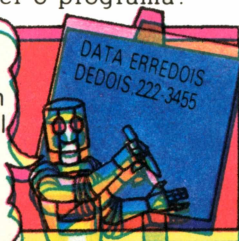
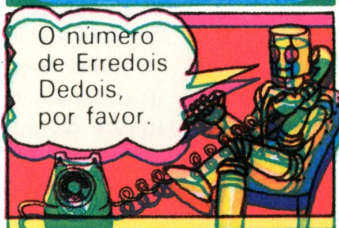
A figura acima mostra o cardápio do Bar Bado. Usando os nomes e os preços dos pratos como dados, escreva um programa que lhe diga o que pode pedir se dispuser de uma certa quantia em dinheiro, como está indicado na tela do lado direito.

Catálogo telefônico

Abaixo estão as instruções para programar um catálogo telefônico. As telas à direita mostram dois exemplos. Você sabe escrever o programa?

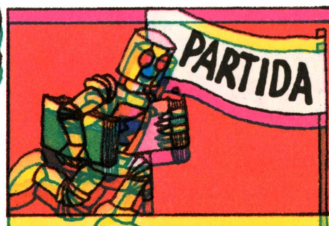
QUAL O NOME DA PESSOA?
?DARTH VADER
NAO ESTA NA LISTA
QUER CONTINUAR? NAO

Guarde os números em uma variável string. Sabe por quê?

DATA ERREDOIS DEDOIS 222-3455

1. Coloque os nomes e os números de telefone de seus amigos em uma instrução DATA.

2. Use PRINT para o computador perguntar o nome da pessoa e INPUT para você responder.

3. Use READ para procurar o nome na lista. Use uma variável para os nomes e outra para os números.

4. Imprima o nome e o número (que deve estar logo depois) ou diga que o nome não consta da lista.

5. Faça o computador perguntar se você quer continuar. Use INPUT para responder.

6. Dependendo da resposta, use RESTORE para voltar ao ponto de partida ou STOP para encerrar o programa.

Uso de matrizes

Os dados também podem ser guardados em matrizes. Você pode pensar em uma matriz como uma variável com vários compartimentos numerados, cada um dos quais pode conter um dado. Para se referir a um elemento em particular, você deve especificar o nome da matriz e o número do compartimento, que é chamado de índice.

Matrizes numéricas

Aqui está uma matriz numérica chamada N. Ela contém seis elementos. Você precisa dizer ao computador qual é o tamanho da matriz, para que ele possa reservar um espaço suficiente na memória. Para isso, utiliza-se a instrução DIM, seguida do nome da matriz e do número de elementos que contém. Isto é chamado de dimensionar uma matriz.

```
*10 DIM ?
 20 FOR K=1 TO 6
 30 READ N(K)
 40 NEXT K
*50 DATA ?
```

Para o TK-83

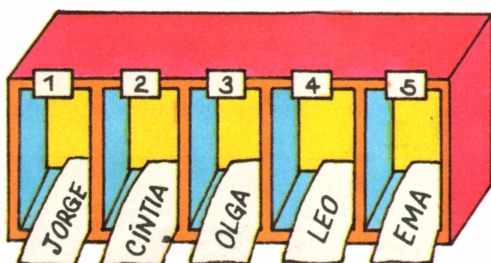
```
*10 DIM?
 20 FOR K=1 TO 6
 30 INPUT N(K)
 40 NEXT K
```

```
N(1)  VALE  1066
N(2)  VALE  1216
N(3)  VALE  1485
N(4)  VALE  1603
N(5)  VALE  1665
N(6)  VALE  1959
```

```
1485   1216
1959   1485
1959   1665
1603   1603
1485   1216
```

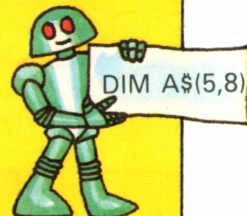
Para guardar os dados em uma matriz, pode-se usar um loop com READ e DATA. Complete o programa da esquerda para guardar todos os dados da figura acima em uma matriz. Se o seu micro é um TK-83, use o programa da direita.

Agora faça um programa para imprimir os dados guardados na matriz. Use PRINT e uma variável como índice da matriz. Na tela da direita, o computador imprime elementos da matriz ao acaso, usando um número aleatório como índice da matriz.

Matrizes string

Esta é uma matriz string (N$). Ela contém cinco elementos, cada um com um nome armazenado. Escreva um programa para guardar dados nesta matriz e depois mostrar os dados na tela.

Micros da família Sinclair

MATRIZ A$
1 GATO
2 PEIXE
3 ELEFANTE
4 RATO
5 BURRO

DIM A$(5,8)

Nos micros da família Sinclair, cada string é guardada em uma linha diferente da matriz e cada caractere de uma string é guardado em um elemento da matriz. Para dimensionar uma matriz, você tem que dizer ao computador quantas strings (linhas) existem e qual o número de caracteres (elementos) da maior string. Coloque os dois números entre parênteses, como no exemplo acima. O computador iguala os comprimentos de todas as strings, acrescentando espaços às outras strings.

Calendário

```
* 10 ?
  20 FOR K=1 TO 12
* 30 ?
  40 NEXT K
  50 PRINT "O MES NUMERO ";
  60 INPUT N
* 70 PRINT M$(?);" TEM ";
* 80 PRINT D(?);" DIAS"

* 90 DATA ?
*100 DATA ?
*110 DATA ?
```

Linha 10: Dimensione as matrizes.

Linha 30: Leia os dados e guarde em duas variáveis, M$ e D.

Linhas 70 e 80: Coloque os índices apropriados para imprimir os dados corretos.

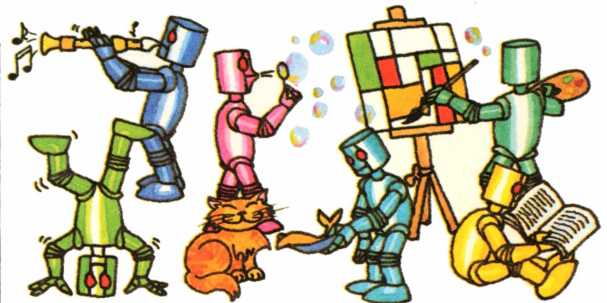
Linhas 90 a 110: Coloque nessas linhas o número de cada mês seguido pelo número de dias correspondente.

Complete o programa acima para que, quando você fornecer o número do mês, o programa escreva o nome do mês e o número de dias que ele tem.

```
QUAL O NOME DO MES?
?JULHO

JULHO TEM 31 DIAS
```

Agora mude o programa para o computador perguntar o nome do mês. Use um loop e IF...THEN para procurar o nome do mês na matriz M$.

Programa de sugestões

Aqui está um programa que pode ser útil para quando você não tem o que fazer. Você escolhe um número e o computador mostra uma sugestão na tela.

Para escrever o programa, você precisa de uma matriz string (I$) com dez sugestões diferentes. Coloque um número em uma variável (N) usando INPUT e use N como índice de I$.

Vinte perguntas

A tela acima mostra um jogo de perguntas e respostas. Para escrever o programa, faça 20 perguntas. Coloque as perguntas em uma matriz e as respostas em outra, na mesma ordem.

Gráfico de números aleatórios

```
10 LET N=0
20 DIM A(10)
30 FOR K=1 TO 10
40 LET A(K)=0
50 NEXT K
60 LET R=INT(RND(1)*10+1)
70 LET A(R)=A(R)+1
80 LET N=N+1
90 IF N<100 THEN GOTO 60
```

O computador dá a cada elemento da matriz A um valor inicial de 0.

Este programa usa uma matriz para guardar os dados de uma tabela. O computador escolhe ao acaso 100 números entre 1 e 10. Cada elemento da matriz (A) contém o número de vezes que cada número foi escolhido. Veja se você consegue escrever um programa para fazer um gráfico como o da tela acima, que mostra uma estrela para cada vez que o número foi escolhido.

Uso de sub-rotinas

Sub-rotina é uma parte de um programa que é usada várias vezes durante a execução do programa. A instrução GOSUB, seguida pelo número da primeira linha da sub-rotina, diz ao computador para saltar para a sub-rotina. O computador executa a sub-rotina até chegar à instrução RETURN. Em seguida, volta para o programa principal, começando pela instrução seguinte a GOSUB.

Rode o programa abaixo para ver como funcionam as instruções GOSUB e RETURN.

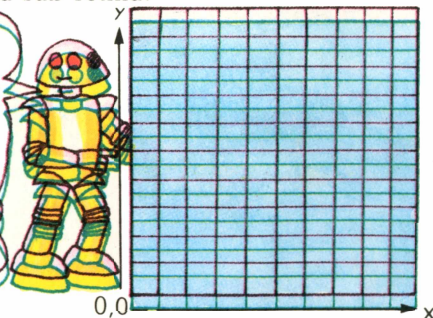
```
  10 PRINT "BOM DIA ";
  20 GOSUB 1010
  30 PRINT "BOM DIA DE NOVO ";
  40 GOSUB 1010
  50 PRINT "E QUE E ISTO?"
  60 GOSUB 2020
  70 STOP
1010 PRINT "ISTO E UMA SUB-ROTINA"
1020 RETURN
2020 PRINT "OUTRA SUB-ROTINA"
2030 RETURN
```

Pesquisa de sorvete	
MANGA	16
ABACAXI	11
COCO	8
MORANGO	1
GRAVIOLA	18

A tabela acima mostra o resultado de uma pesquisa de popularidade envolvendo seis sabores de sorvete. Escreva um programa para mostrar os dados como na tela à direita. Use um loop para guardar os dados em duas matrizes e, dentro do loop, mande o computador a uma sub-rotina para imprimir uma linha de cada vez.

Afunde o submarino

O programa abaixo é para um jogo chamado "Afunde o Submarino". Um submarino inimigo está escondido em algum lugar de uma rede 10 x 10. O computador determina sua posição escolhendo dois números aleatórios que funcionam como coordenadas. Você tem quatro chances de localizar o submarino adivinhando as coordenadas X (horizontal) e Y (vertical). Se errar, o computador vai para uma sub-rotina que lhe diz para onde deve ir na tentativa seguinte. Para poder rodar o programa, você precisa escrever a sub-rotina.

```
 10 CLS
 20 LET N=0
 30 LET X=INT(RND(1)*10+1)
 40 LET Y=INT(RND(1)*10+1)
 50 LET N=N+1
 60 PRINT "PALPITE NO.";N;" ";
 70 INPUT A,B
 80 IF A=X AND B=Y THEN GOTO 170
 90 GOSUB 200
100 PRINT
110 IF N<=4 THEN GOTO 50
120 PRINT "ACABARAM OS TIROS"
130 PRINT "QUER JOGAR DE NOVO?"
140 INPUT R$
150 IF LEFT$(R$,1)="S" THEN GOTO 10
160 STOP
170 PRINT "VOCE AFUNDOU O SUBMARINO"
180 PRINT "EM ";N;" TENTATIVAS"
190 GOTO 130
```

Como escrever a sub-rotina

Você precisa de várias linhas de IF...THEN para comparar seu palpite (A,B) com a posição do submarino (X,Y) e imprimir uma mensagem na tela. Assim, por exemplo, se B é menor que Y, você deve escrever N (norte) etc.

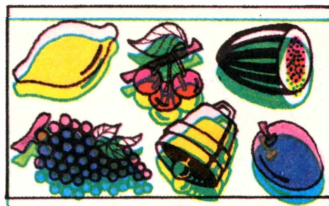
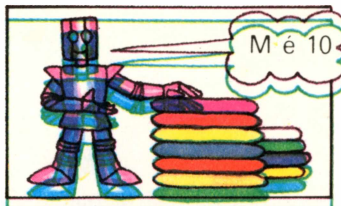
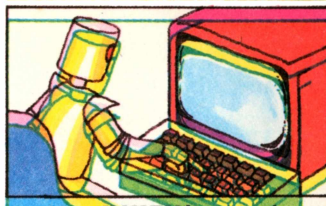
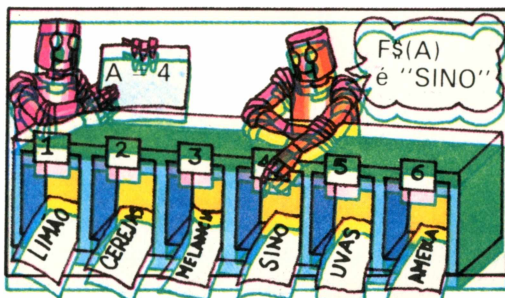
```
PALPITE NO.1 ?3,2
ERROU
VA PARA NE
```

```
PALPITE NO.2 ?3,3
ERROU
VA PARA E
```

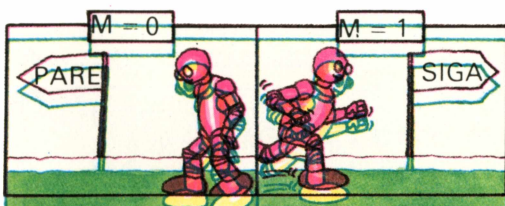
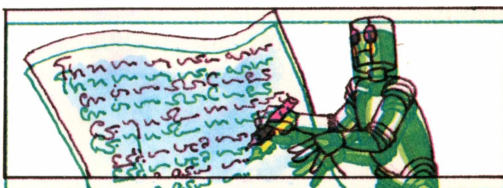
```
PALPITE NO.3 ?5,3
ERROU
VA PARA O
```

Programe uma máquina caça-níqueis

Você é capaz de escrever um programa para fazer o seu micro funcionar como uma máquina caça-níqueis? A tela à direita mostra um exemplo. Você começa com 10 moedas e gasta uma cada vez que joga. Aperte qualquer tecla para começar e o computador imprimirá o resultado. As instruções abaixo o ajudarão a escrever o programa.

```
VOCE TEM 10 MOEDAS
APERTE QUALQUER TECLA:P
SINO  SINO  LIMAO
2 IGUAIS
VOCE GANHOU 2 MOEDAS
AGORA TEM 11 MOEDAS
APERTE QUALQUER TECLA:B
```

1. Dimensione uma matriz **F$** para guardar os nomes dos seis objetos acima (limão, cerejas, ameixa, melancia, uvas e sino). Guarde os nomes usando **READ** e **DATA** (ou **INPUT**).

2. Apague a tela e use uma variável M para guardar o número de moedas. Inicialmente, faça M = 10.

3. Use **INKEY$** para fazer o computador esperar que o jogador aperte uma tecla qualquer. Subtraia 1 de M (custa uma moeda para jogar).

4. Faça o computador escolher um número aleatório entre um e seis e guarde-o em A. Use A como índice de **F$** para escolher um objeto e guarde-o em **A$**. Faça isso mais duas vezes e guarde os resultados em **B$** e **C$**.

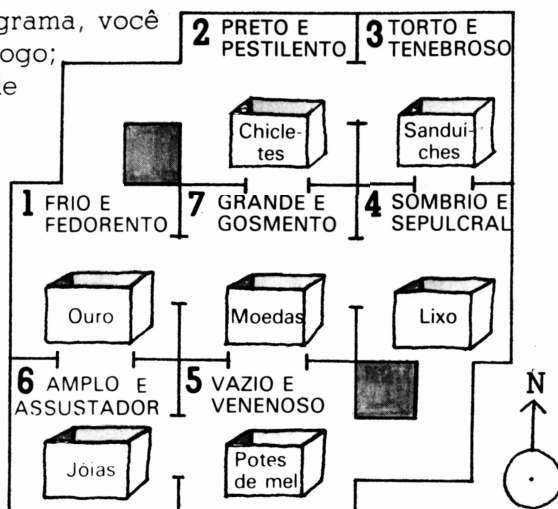
3 CEREJAS:	GANHA 20 MOEDAS
3 IGUAIS:	GANHA 5 MOEDAS
2 IGUAIS:	GANHA 2 MOEDAS

5. Agora você deve escrever o resultado na tela e fazer o computador decidir se você ganhou alguma coisa ou não, de acordo com a tabela acima. Para cada tipo de resultado, mande o computador para uma sub-rotina diferente.

6. Verifique se o jogador ainda tem pelo menos uma moeda. Se as moedas acabaram (M = 0), pare o programa; caso contrário, continue o jogo.

7. Se você usou **READ** no começo do programa, não se esqueça de colocar as linhas de **DATA** no final.

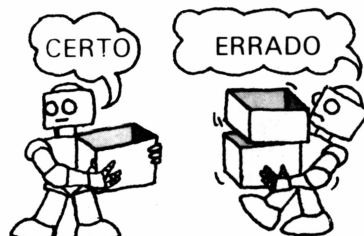
Programe uma caça ao tesouro

Nas próximas páginas, você encontrará as instruções para escrever um programa de caça ao tesouro, no qual terá que passar por um labirinto de sete quartos, recolhendo objetos . O programa é bastante complexo; procure seguir as instruções à risca e verifique cada etapa antes de prosseguir. Se ficar "empacado" em uma parte do programa, veja a resposta apenas para esta parte e continue a escrever o resto do programa. Para poder escrever o programa, você deve ter uma boa idéia de como é o jogo; leia o parágrafo que se segue antes de começar.

Regras do jogo

O mapa à direita mostra um labirinto de sete quartos. Cada um tem um nome diferente e contém uma caixa cheia de objetos. As informações do mapa estão guardadas na memória do computador, mas o mapa não aparece na tela. Isto quer dizer que você tem que descobrir a localização dos quartos enquanto estiver jogando o jogo. Partindo de um quarto escolhido ao acaso, o objetivo é reunir todas as caixas no mesmo quarto em um número limitado de movimentos.

```
ESTAS SAO AS PALAVRAS QUE O COMPUTADOR COMPREENDE

N,S,L,O: ANDAR PARA O NORTE,SUL,LESTE OU OESTE
PEGAR  : PEGAR CAIXA
LARGAR : LARGAR CAIXA
LOCAL  : MOSTRA A LOCALIZACAO DAS CAIXAS
AJUDA  : MOSTRA AS REGRAS DO JOGO
```

A tela acima mostra os comandos que você pode usar e para que servem. Você só pode dar um comando em cada jogada.

Para tornar o jogo mais difícil, você só pode carregar uma caixa de cada vez.

Exemplo

```
1 VOCE ESTA NO QUARTO 4
  ELE E SOMBRIO E SEPULCRAL
  ELE CONTEM LIXO

  QUE VAI FAZER?
  ? PEGAR

  OK, VOCE ESTA CARREGANDO LIXO
```

```
2
  VOCE AINDA ESTA NO QUARTO 4
  QUE VAI FAZER?
  ? O

  OK
```

```
3 VOCE ESTA NO QUARTO 7
  ELE E GRANDE E GOSMENTO
  ELE CONTEM MOEDAS

  QUE VAI FAZER?
  ? LARGAR

  OK, LIXO NO QUARTO 7
```

```
4
  VOCE AINDA ESTA NO QUARTO 7
  QUE VAI FAZER?
  ? S

  OK
```

Como escrever o programa

O diagrama à direita é um fluxograma. Ele mostra a estrutura do programa. As instruções abaixo e nas páginas seguintes o ajudarão a escrevê-lo. Não se preocupe com a seqüência em que vai fazer as diferentes etapas. Se usar a numeração das linhas sugerida, o programa será montado na ordem correta.

1 Definição das matrizes e leitura dos dados (linhas 100-260)

Você precisa colocar todas as informações que aparecem no mapa da página anterior na memória do computador. Para isso, vai ter que definir várias matrizes.

Matrizes, N,L,S e O

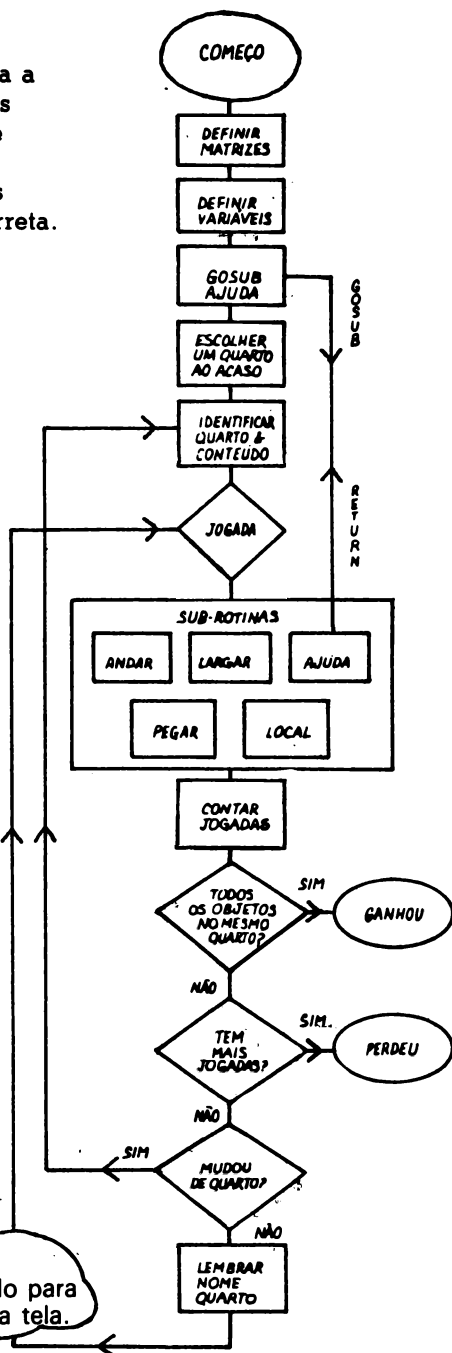
Em primeiro lugar, você precisa de quatro matrizes chamadas N,L,S e O. Os dados guardados nessas matrizes dizem ao computador que quarto está ao norte, leste, sul e oeste de cada quarto do labirinto. Dimensione essas matrizes na primeira linha (100)*. Cada matriz tem sete elementos.

	N	L	S	O
1	2	7	6	0
2	0	3	7	1
3				
4				
5				
6				
7				

Faça uma tabela como esta para colocar os dados. Os números na primeira coluna são os índices das matrizes e representam os números dos quartos. Para cada linha da tabela, consulte o mapa dos quartos e coloque o número do quarto que está ao norte, leste, sul e oeste do quarto indicado na primeira coluna. As primeiras duas linhas já estão prontas.

Leitura dos dados

```
110 FOR K=1 TO 7
120 READ N(K),L(K),S(K),O(K)
130 NEXT K
2000 DATA 2,7,6,0
```

Você precisa de um loop como o que aparece acima para guardar os dados nas matrizes. Copie os dados da tabela em sete linhas de DATA. A primeira é a linha 2000 acima.

*Se o seu micro é um TK-83, você só pode dimensionar uma matriz por linha, de modo que é melhor começar na linha 10.

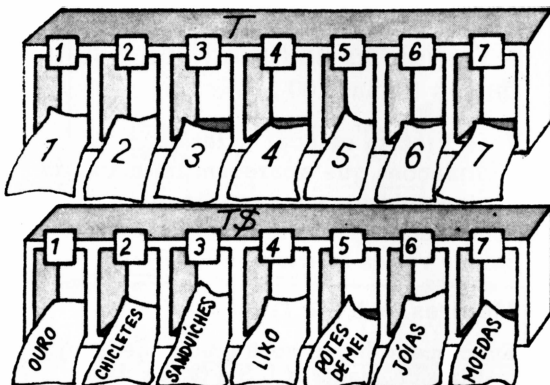
Matriz D$

Leia os dados usando outro loop.

```
2100 DATA FRIO E FEDORENTO, PRETO E PESTILENTO
```

Agora você precisa de uma matriz chamada D$ para guardar as descrições dos quartos. Dimensione-a no começo do programa e escreva os dados como no exemplo acima.

Matrizes T$ e T

Agora você precisa de mais duas matrizes, T$ para guardar os nomes dos objetos e T para guardar sua localização. O número guardado em cada elemento de T é o número do quarto onde está o objeto cujo nome está guardado no mesmo elemento de T$. Por exemplo: no começo do jogo, T(2) é 2 e este é o quarto onde está guardado o objeto T$(2) (chicletes). Durante o jogo, os números guardados em T mudam.

```
2200 DATA OURO,1,CHICLETES,2
2210 DATA SANDUICHES,3,LIXO,4
```

Variáveis

M	C	F	WXY
Número de jogadas já feitas.	Mostra se o jogador está carregando alguma coisa.	Uma "bandeira". Depois você verá para que serve.	Variáveis para guardar dados temporários.

A figura acima mostra os nomes das variáveis que você vai usar no programa. Começando na linha 300, dê a todas elas um valor inicial de 0.

2 Sub-rotina Ajuda (linhas 1000-1120)

```
EXISTEM 7 QUARTOS NO LABIRINTO
E UM OBJETO EM CADA QUARTO. VOCE
DEVE REUNIR TODOS OS OBJETOS
NO MESMO QUARTO.
```

Você precisa de uma sub-rotina para explicar as regras do jogo. Coloque uma instrução GOSUB na linha 300 e comece a sub-rotina na linha 1000. Escreva as instruções à esquerda e depois os comandos que o computador compreende (página 28).

3 Escolha de um quarto ao acaso (linha 350)

O número do quarto onde está o jogador é guardado na variável R. Para escolher o quarto inicial, faça o computador escolher um número ao acaso entre 1 e 7 e guarde-o em R na linha 350.

4 Identificação do quarto e seu conteúdo (linhas 400-470)

```
VOCE ESTA NO QUARTO 7
ELE E GRANDE E GOSMENTO
ELE CONTEM MOEDAS
```

```
VOCE ESTA NO QUARTO 4
ELE E SOMBRIO E SEPULCRAL
ELE CONTEM NADA
```

```
VOCE ESTA NO QUARTO 1
ELE E FRIO E FEDORENTO
ELE CONTEM OURO
```

Antes de tudo, diga ao jogador em que quarto está. Depois, descreva o quarto usando R como índice de D$. As telas acima mostram alguns exemplos.

Identificação do conteúdo do quarto

```
430 FOR K=1 TO 7
440 IF T(K)=R THEN PRINT T$(K):
LET F=1
450 NEXT K
460 IF F=0 THEN PRINT "NADA"
470 LET F=0
```

Para descobrir o conteúdo do quarto R, você tem que consultar a matriz T, onde está guardado o número dos quartos onde estão os objetos. Para isso, use o loop acima. A instrução IF...THEN é para verificar se algum dos números em T é igual a R. Em caso afirmativo, o computador imprime o nome do objeto e muda a variável F para 1.

A variável "bandeira"

A variável F diz ao computador se existe algum objeto no quarto depois que ele sai do loop. Ela funciona como uma bandeira. Quando um objeto é encontrado, a bandeira é levantada (F = 1). Se nenhum objeto é encontrado, a bandeira continua baixada (F = 0). Lembre-se de fazer F = 0 na linha 470 para usar a bandeira de novo.

5 Jogada (linhas 500-560)

Agora pergunte ao jogador o que vai fazer e use uma linha de INPUT que permita ao jogador guardar um comando (como PEGAR, LARGAR etc.) na variável A$. Escreva cinco linhas de GOSUB que mandam o computador para sub-rotinas diferentes para cada comando.

6 Sub-rotina Andar

(linhas 1200-1260)

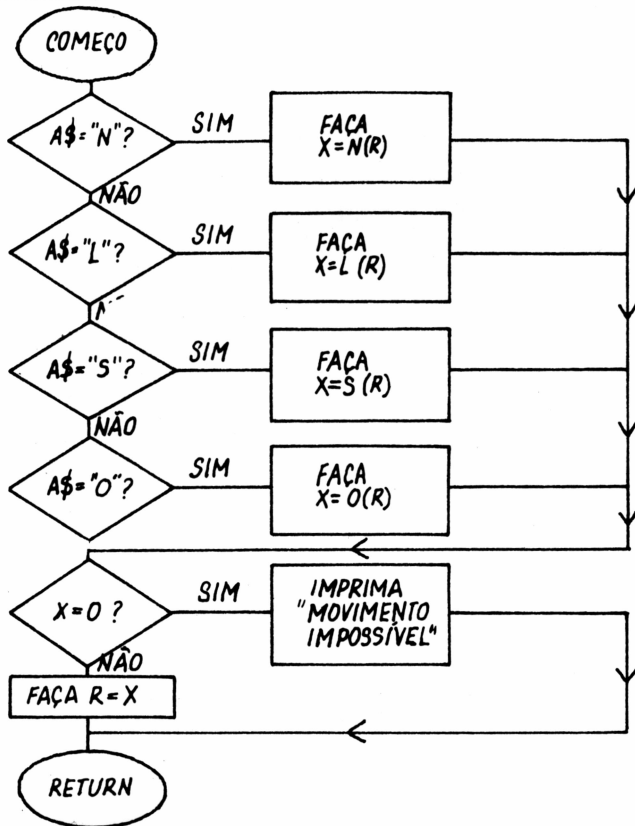
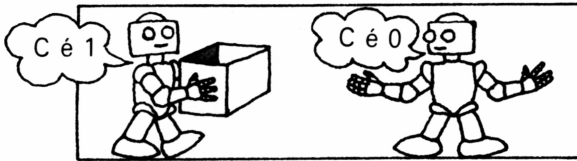
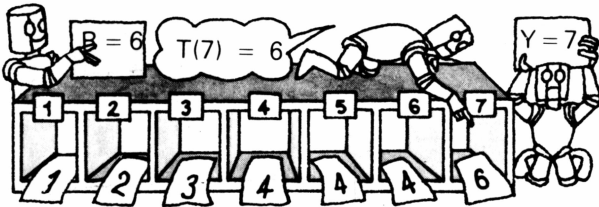
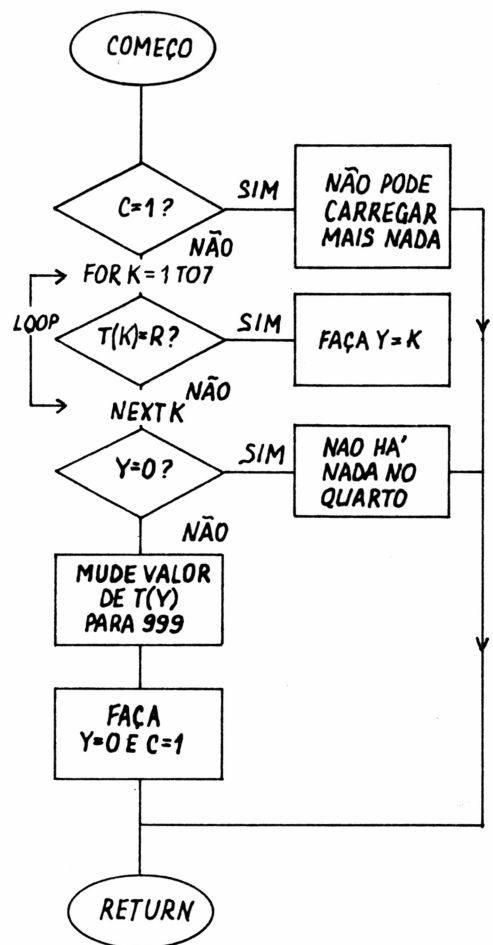
Escreva a sub-rotina Andar com o auxílio do fluxograma ao lado.

1. Para andar, o jogador escreve N,L,S ou O. Verifique qual foi a letra usando quatro instruções IF...THEN.

2. Para descobrir o número do novo quarto, use R como índice da matriz correspondente (N,L,S ou O) e guarde o resultado na variável temporária X.

3. Agora teste X para ver se o jogador pode andar nessa direção. Se X é 0, não existe nenhum quarto nessa direção; você deve imprimir uma mensagem e usar um GOTO para mandar o computador para a instrução RETURN no fim da sub-rotina.

4. Se X não é 0, coloque o número do novo quarto em R fazendo R = X.

5. Coloque RETURN no final da sub-rotina.

7 Sub-rotina Pegar (linhas 1300-1380)

1. O jogador só pode carregar uma caixa de cada vez, de modo que primeiro é preciso ver se está carregando alguma coisa. Para isso, teste o valor da variável C, que passa para 1 quando o jogador pega uma caixa. Se C é 1, imprima uma mensagem e mande o computador para RETURN.

2. Para verificar se existe uma caixa no quarto, programe um loop usando IF...THEN para examinar a matriz T. Se algum número em T é igual ao número do quarto (R), você deve guardar o número do índice de T correspondente (representado por K) em uma variável temporária chamada Y. O computador pode encontrar vários números iguais a R. Nesse caso, quando o loop terminar, Y conterá o índice do último elemento encontrado.

3. Se Y é 0, então não há nenhuma caixa no quarto, de modo que você deve mandar o computador para RETURN; caso contrário, o número guardado em Y diz ao computador que caixa deve pegar. Mude o valor de T (Y) para 999, um número que mostra que a caixa está sendo carregada. Faça C = 1 e Y = 0 e coloque RETURN no final da sub-rotina.

8 Sub-rotina Largar (linhas 1400-1460)

1. Verifique se o jogador está carregando alguma coisa, testando o valor de C.

COMEÇO

O JOGADOR ESTÁ CARREGANDO ALGUMA COISA?

NÃO → NADA PARA LARGAR

SIM

FOR K = 1 TO 7

LOOP

T(K)=999?

SIM → FAÇA T(K)=R

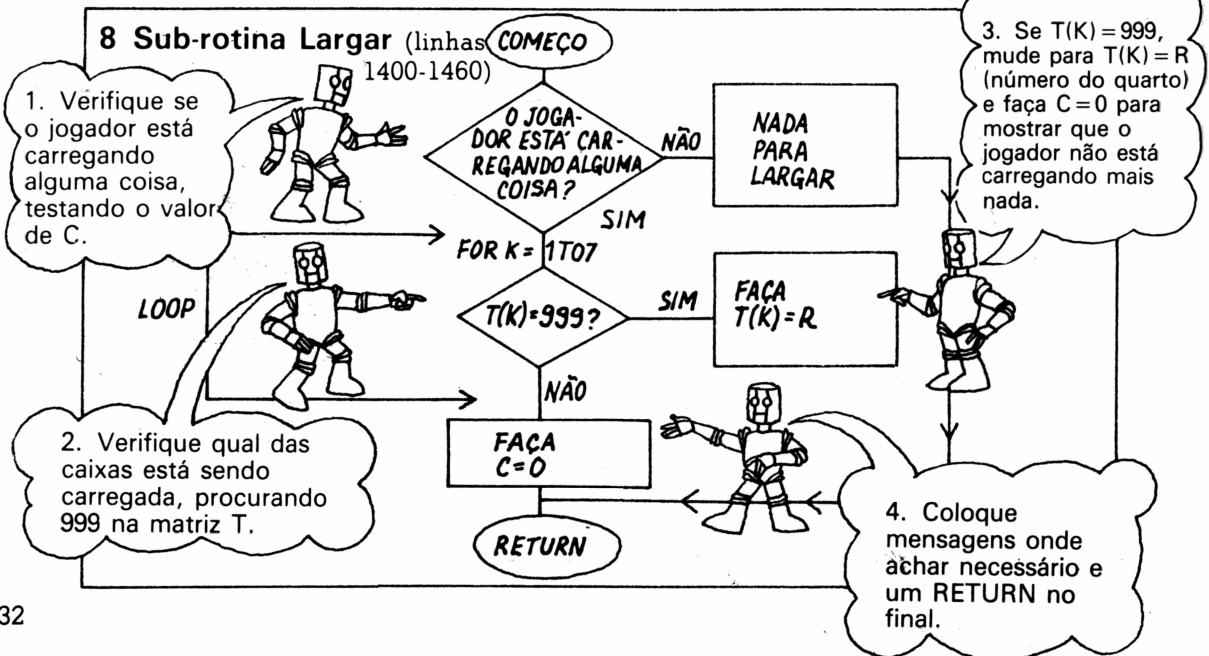
NÃO

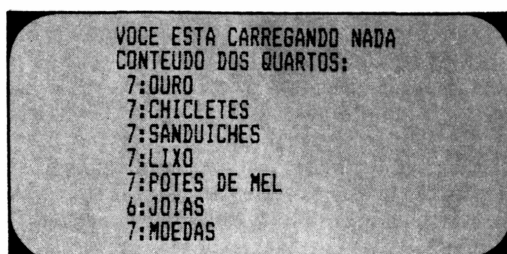
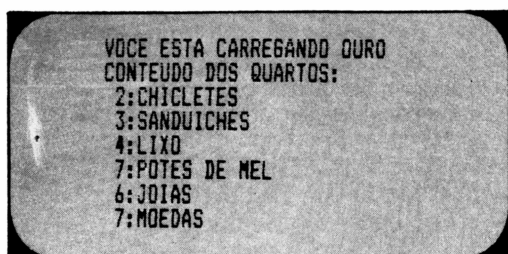
FAÇA C=0

RETURN

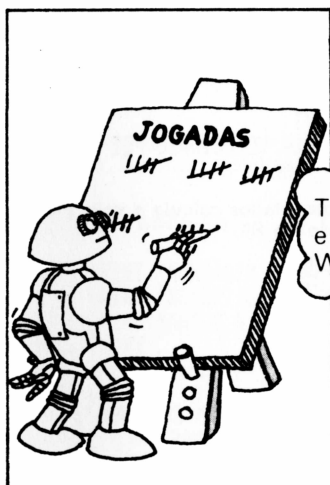
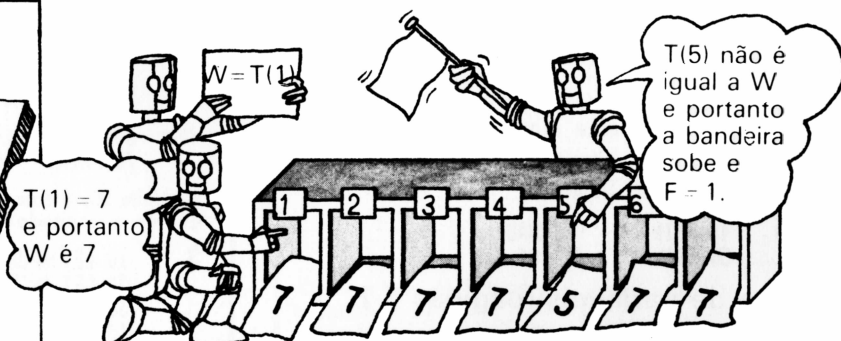
2. Verifique qual das caixas está sendo carregada, procurando 999 na matriz T.

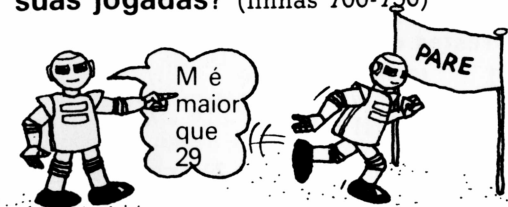
3. Se T(K) = 999, mude para T(K) = R (número do quarto) e faça C = 0 para mostrar que o jogador não está carregando mais nada.

4. Coloque mensagens onde achar necessário e um RETURN no final.

9 Sub-rotina Local (linhas 1500-1590)

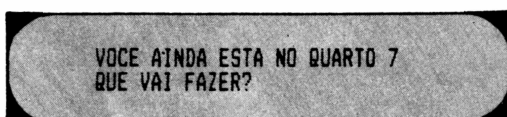
Esta sub-rotina fornece ao jogador a localização de todas as caixas. Use as telas abaixo como modelo. Prepare um fluxograma antes de escrever o programa.

Primeiro, teste o valor de C para ver se o jogador está carregando alguma coisa. Se está, descubra qual o elemento da matriz T é igual a 999 e imprima o nome da caixa escolhendo o elemento da matriz T$ com o mesmo índice. Para imprimir o conteúdo dos quartos, utilize um segundo loop.

10 Contagem das jogadas (linha 600)

11 Todas as caixas estão no mesmo quarto? (linhas 610-690)

Na linha 600, some 1 à variável M para atualizar o número de jogadas já feitas.

Se todas as caixas estão no mesmo quarto, o jogador ganhou. Teste os números da matriz T; se forem todos iguais, as caixas estão no mesmo quarto. Para isso, tome um elemento de T, T(1), por exemplo, e guarde-o em uma variável temporária chamada W. Compare W com todos os outros números guardados em T. Use uma "bandeira" para indicar se são todos iguais ou não.

12 O jogador usou todas as suas jogadas? (linhas 700-750)

Escolha a quantas jogadas o jogador tem direito. Se M for maior que o número escolhido, o jogador perdeu. Nesse caso, imprima uma mensagem e pare o programa.

13 O jogador mudou de quarto? (linhas 800-850)

```
VOCE AINDA ESTA NO QUARTO 7
QUE VAI FAZER?
```

Se o jogador colocou N,L,S ou O na variável **A$**, mande o computador de volta para a linha 400 para identificar o novo quarto e seu conteúdo; caso contrário, imprima a antiga localização e mande o computador para a linha 500.

Respostas dos problemas

Nas próximas páginas, você encontrará as soluções dos problemas propostos neste livro. Os programas foram escritos em um dialeto específico de BASIC; talvez você tenha que mudar algumas instruções não-padronizadas, como RND e CLS. Se um dos programas listados aqui não funcionar, verifique primeiro se o seu micro usa todas as instruções de BASIC do programa. Existe uma tabela de conversão na página 47 para ajudá-lo. As linhas de programas que precisam ser modificadas para o programa funcionar nos micros da família Sinclair estão marcadas com um △ e as linhas corretas aparecem logo abaixo. As linhas que precisam ser mudadas apenas para o TK-83 estão marcadas com um ▲.
Pode ser que algumas das respostas sejam diferentes das dos programas que você escreveu. Se o seu programa funcionar corretamente, tudo bem. Entretanto, examine a resposta e compare-a com a sua, para ver qual a que contém o menor número de instruções.

Exercícios com PRINT (páginas 4-5)

Alguns quebra-cabeças

1
```
40 PRINT "BOM DIA ";
50 PRINT A$
```

O ponto-e-vírgula faz o computador ficar na mesma linha para imprimir o conteúdo de **A$**.

2
```
40 PRINT "BOM DIA",A$
50 apague esta linha
```

A vírgula faz o computador deixar alguns espaços antes de imprimir o conteúdo de **A$**.

3
```
40 PRINT TAB(6);"BOM DIA ";A$
50 apague esta linha
```

Alguns computadores não precisam de ponto-e-vírgula depois de TAB.

4
```
40 PRINT "     TCHAU"
50 PRINT "     ";A$
```

Uso de variáveis (páginas 6-7)

Como escolher os nomes das variáveis

Os seguintes nomes contêm palavras de BASIC e não podem ser usados como nomes de variáveis: LETRA$(LET); RUN$(RUN). No TK-83, os nomes das variáveis strings só podem ter uma letra, de modo que nesse micro **PULGA$** também não seria permitido.

Teste com PRINT

```
  10 LET A=66
  20 LET B=77
▲ 30 LET RR$="SAPOS-MARTELOS"
  40 PRINT A;" ";RR$
  50 PRINT "COMERAM ";B;" MOSCAS"
```

Um espaço entre aspas faz o computador pular um espaço.

Analisando um programa

```
45 PRINT "C= ";C
55 PRINT "D= ";D
```

Centígrados em Fahrenheit

```
40 LET C=9/B
50 LET D=C*A
60 LET R=D+F
```

Substitua as linhas 40 e 60 pelas linhas acima.

Idéias para programas

1
```
10 LET F$="PF"
20 PRINT "ESCREVA A PALAVRA "
30 INPUT P$
40 PRINT "A PALAVRA EM PFIANO E ";
50 PRINT F$;P$
```

A linha 50 faz o computador imprimir PF, seguido pela palavra que está guardada em **P$**.

2
```
10 PRINT "QUAL A DISTANCIA ";
20 INPUT D
30 PRINT "QUANTO TEMPO LEVOU ";
40 INPUT T
50 LET S=D/T
60 PRINT "SUA VELOCIDADE FOI ";S;
70 PRINT " KM/H"
```

Na linha 50, o computador calcula a velocidade e guarda o resultado em S*.

3
```
 10 LET A=30
 20 LET B=20
 30 LET S=35
 40 LET T=S/B
 50 LET C=A*T
 60 LET D=C+S
 70 PRINT "ELES DEVEM COMPRAR ";
 80 PRINT D;" SALSICHAS."
 90 PRINT "ELES VAO LEVAR ";
100 PRINT T;" HORAS"
```

A e B são as salsichas que os robôs comem por hora. S são as salsichas que o robô 2 quer comer e T é o tempo que o robô 2 leva para comer suas salsichas (S). A linha 50 calcula o número de salsichas que o robô 1 come no mesmo tempo (T). D é o número total de salsichas.

Repetições (páginas 8-9)

Testes com loops

1
```
10 FOR J=1 TO 100
20 PRINT "BOM DIA ";
30 NEXT J
```

O ponto-e-vírgula na linha 20 faz o computador ficar na mesma linha para imprimir a palavra seguinte.

2
```
10 FOR J=1 TO 25
20 PRINT TAB(15);"BOM DIA"
30 NEXT J
```

O programa faz o computador imprimir uma coluna de 25 BOM DIA, a 15 espaços da esquerda da tela.

*Em alguns computadores, não é preciso colocar ponto-e-vírgula no final das linhas de PRINT antes de INPUT.

3 A linha 20 está errada porque interfere com o contador do loop. Cada vez que o computador passa pelo loop, a linha 20 faz o valor de I mudar de 1 para 0.

Testes com STEP

1

```
10 FOR I=25 TO 1 STEP-1
20 PRINT TAB(I);"BOM DIA"
30 NEXT I
```

Use a variável do loop (I) como número do TAB. Certifique-se de que I não é maior que a largura da sua tela. Cada vez que o loop é repetido, I diminui de 1.

2

```
10 FOR L=5 TO 0 STEP-1
20 PRINT TAB(5);L;TAB(10);L*L
30 NEXT L
```

STEP-1 faz a variável do loop (L) contar para trás de 5 até 0. Cada vez que o loop é repetido, a linha 20 imprime o valor de L e do quadrado de L.

3

```
10 PRINT "INICIO ";
20 INPUT A
30 PRINT "FIM ";
40 INPUT B
50 PRINT "INCREMENTO ";
60 INPUT C
70 FOR J=A TO B STEP C
80 PRINT J;" ";
90 NEXT J
```

O loop nas linhas 70 a 90 usa A como valor inicial do contador, B como valor final e C como incremento.

Mensagem secreta

1

```
10 PRINT "MENSAGEM SECRETA"
20 PRINT "MEMORIZE EM 5 SEGUNDOS"
30 PRINT "ANTES QUE DESAPARECA"
40 PRINT
50 PRINT "ENCONTRE AGENTE X NO AEROPORTO"
60 FOR I=1 TO 1000 ]—— Loop de espera
70 NEXT I          ]
80 CLS ——————————— Apaga a tela
```

Desenhando na tela

2

```
50 PRINT TAB(A-K);"*";
60 PRINT TAB(A+K);"*"
70 NEXT K
```

A é o vértice do triângulo. Para imprimir o lado esquerdo você subtrai K e A e para imprimir o direito você soma K a A.

3

```
30 Apague esta linha.
40 FOR K=9 TO 1 STEP-1
80 PRINT TAB(A);"*"
```

Mude as linhas acima para imprimir a figura de cabeça para baixo. Para fazer as outras figuras da página 9, use o mesmo programa com diferentes números em TAB.

Problemas com loops (páginas 10-11)

Loop para fazer o programa ficar mais lento

Você pode acrescentar o seguinte loop interno:

```
45 FOR K=1 TO 1000
46 NEXT K
```

Erros em loops

Às linhas 50 e 70 estão erradas. Devem ser assim:

```
50 NEXT K
70 NEXT I
```

Contador binário

Para fazer um programa capaz de contar até 11111111, você precisa de mais quatro loops (E,F,G,H). Você precisa também renumerar as linhas do programa e alterar as linhas de impressão, assim:

```
PRINT H+G*2+F*4+E*8+D*16+C*32+B*64+
A*128;" = ";
PRINT A;B;C;D;E;F;G;H
```

Mensagem que pisca

```
10 FOR J=1 TO 10
20 CLS
30 FOR K=1 TO 1000:NEXT K
40 PRINT TAB(10);"PERIGO"
50 PRINT TAB(5);"ATAQUE ESPACIAL"
60 FOR K=1 TO 1000:NEXT K
70 NEXT J
```

Aqui está um programa para imprimir uma mensagem que pisca. Talvez você tenha que mudar o tamanho do loop de espera e os números de TAB, de acordo com a velocidade e o tamanho de tela do seu micro. Se o seu micro não aceita linhas com mais de uma instrução, coloque NEXT K em uma linha adicional (35).

Relógio digital

```
10 FOR J=0 TO 59            ┐ Loop dos minutos
20 FOR K=0 TO 59        ┐   │
30 PRINT J;":";K        │ Loop dos segundos
40 FOR L=1 TO 500:NEXT L│   │
50 CLS                  │   │
60 NEXT K               ┘   │
70 NEXT J                   ┘
```

Ajuste o loop de K para uma demora de um segundo.

Decolagem de um foguete

```
   10 CLS
▲  20 FOR I=1 TO 20
   30 PRINT
   40 NEXT I
   50 PRINT "   *   "
   60 PRINT "  ***  "
   70 PRINT "  ***  "
   80 PRINT "  ***  "
   90 PRINT " ** ** "
  100 FOR J=1 TO 25
▲ 110 PRINT
  120 FOR K=1 TO 1000 ]—— Loop de espera
  130 NEXT K          ]
  140 NEXT J
```

O loop das linhas 20-40 faz o computador imprimir linhas vazias, de modo que o foguete aparece na parte de baixo da tela. As linhas 50-90 desenham o foguete. O loop nas linhas 100-140 deve ser repetido um número de vezes igual ao número de linhas da sua tela. Cada vez que o loop é repetido, o computador imprime uma linha em branco e o foguete sobe uma linha na tela.

Micros da família Sinclair

Para rodar o programa no TK-85, aperte ENTER quando a mensagem scroll aparecer na tela.

▲Para o TK-83, faça as seguintes mudanças:

```
20 FOR I=1 TO 17
110 SCROLL
```

Programa de ginástica

```
10 FOR J=1 TO 25 STEP 2
20 CLS
30 PRINT
40 PRINT TAB(J);" O "
50 PRINT TAB(J);"<O>"
60 PRINT TAB(J);"/ \"
70 FOR K=1 TO 1000
80 NEXT K
90 CLS
100 PRINT TAB(J+1);"<O>"
110 PRINT TAB(J+1);" O "
120 PRINT TAB(J+1);"/ \"
130 FOR K=1 TO 1000
140 NEXT K
150 NEXT J
```

O número de repetições do loop de J deve ser igual ao número de colunas da sua tela. As linhas 40 a 60 mostram o ginasta na primeira posição. As linhas 70-80 e 130-140 são loops de espera. As linhas 100 a 120 mostram o ginasta na segunda posição.

Exercícios com IF...THEN (páginas 12-13)

Tabuada do 13

```
55 IF B<>J*A THEN PRINT "ERRADO. ";
J;" X ";A;" = ";J*A
```

Acrescente esta linha para o computador dizer que a resposta está errada e fornecer a resposta certa.

Senha

```
50 IF P$<>S$ THEN PRINT "ERRADO.
VA EMBORA"
```

Acrescente a linha acima para completar o programa da senha.

```
10 LET S$="SALSICHAS"
20 LET N=007
30 PRINT "SENHA,POR FAVOR ";
40 INPUT P$
50 PRINT "NUMERO SECRETO ";
60 INPUT NS
70 IF P$<>S$ OR NS<>N THEN PRINT
"ERRADO. VA EMBORA"
80 IF P$=S$ AND NS=N THEN PRINT
"OK. PODE ENTRAR"
```

No programa acima, o computador também pede um número secreto.

Calculadora

```
10 PRINT "DIGA UM NUMERO ";
20 INPUT X
30 PRINT "DIGA OUTRO ";
40 INPUT Y
50 PRINT "VOCE QUER :"
60 PRINT "SOMAR, SUBTRAIR, DIVIDIR
70 PRINT "OU MULTIPLICAR?"
80 INPUT A$
90 IF A$="SOMAR" THEN LET A=X+Y
100 IF A$="SUBTRAIR" THEN LET A=X-Y
110 IF A$="DIVIDIR" THEN LET A=X/Y
120 IF A$="MULTIPLICAR" THEN LET A=X*Y
130 PRINT "A RESPOSTA E ";A
```

A operação que o computador executa depende da palavra que você coloca em A$.

Jogos de adivinhação

```
45 IF Y<X THEN PRINT "MUITO GRANDE"
46 IF Y>X THEN PRINT "MUITO PEQUENO"
```

Acrescente as linhas acima para o computador dizer se o seu palpite é muito grande ou muito pequeno.

```
5 LET N=0
55 LET N=N+1
56 IF N>5 THEN STOP
```

Acrescente as linhas acima para contar o número de tentativas e parar o programa depois de cinco.

Jogo de palavras

```
10 PRINT "DIGA UMA PALAVRA ";
20 INPUT W$
30 PRINT "DIGA UMA PISTA ";
40 INPUT C$
50 CLS
60 PRINT "PISTA"
70 PRINT C$
80 INPUT G$
90 IF G$=W$ THEN GOTO 120
100 PRINT "NAO"
110 GOTO 80
120 PRINT "SIM"
```

Nas linhas 10 a 40 o computador pede uma palavra e uma pista. As linhas 50 e 60 limpam a tela e imprimem a pista. A linha 110 manda o computador de volta à linha 80 para outro palpite.

Corrida de cavalos

Aqui estão as instruções IF...THEN completas:

```
80 IF G1=H1 AND G2=H2 THEN GOTO 160
90 IF G1=H2 OR G2=H1 THEN GOTO 150
100 IF (G1=H1 AND G2<>H2) OR (G2=H2
AND G1<>H1) THEN GOTO 140
120 IF N=4 THEN GOTO 170
```

Linha 80: Você deve mandar o computador para a linha 160 se os dois palpites estiverem certos.
Linha 90: Se o cavalo está certo mas a colocação está errada, mande o computador para a linha 150.
Linha 100: Se um dos palpites está certo, vá para a linha 140.
Linha 120: Depois de quatro tentativas, mande o computador para a linha 170.

Idéias para melhorar o jogo

```
15 LET S=0
140 PRINT "UM PALPITE CERTO":LET
S=2:GOTO 120
160 PRINT "CERTO":LET S=4
180 PRINT "VOCE FEZ ";S;" PONTOS"
```

Acrescentando as linhas acima ao programa, você ganha quatro pontos se acertar os dois cavalos e dois se acertar apenas um. A variável S é usada para guardar a contagem.

```
190 PRINT "QUER JOGAR DE NOVO? (S/N)"
200 INPUT Z$
210 IF Z$="S" THEN GOTO 10
220 IF Z$="N" THEN STOP
```

Para dar ao jogador a oportunidade de jogar de novo, acrescente as linhas acima.

Números aleatórios (páginas 14-15)

Número aleatório entre 10 e 20

```
INT(RND(1)*11+10)
```

Para obter um número aleatório entre 10 e 20, multiplique o número de números que deseja (11) e depois some o primeiro da lista (10). Use a instrução RND apropriada para o seu micro.

Quebra-cabeças

```
20 LET X=INT(RND(1)*20+1)
```

Mude a linha 20 do jogo de adivinhação de números para fazer o computador escolher um número ao acaso entre 1 e 20.

Seqüência de números

Acrescente as linhas abaixo ao programa:

```
  5 PRINT "DESCUBRA O NUMERO"
  7 PRINT "SEGUINTE NESTA"
  9 PRINT "SEQUENCIA"
 10 LET X=INT(RND(1)*10+1)
 20 LET Y=INT(RND(1)*10+1)
 60 LET A=X+4*Y
 70 INPUT N
 80 IF N=A THEN PRINT "CERTO"
 90 IF N<>A THEN PRINT "ERRADO. E ";A
100 GOTO 5
```

A linha 60 calcula o número seguinte da seqüência.

Seqüência aleatória

```
24 LET R=INT(RND(1)*3+1)
25 IF R=1 THEN GOTO 30
26 IF R=2 THEN GOTO 40
27 IF R=3 THEN GOTO 50
```

Para escolher uma seqüência ao acaso, faça o computador escolher um número aleatório entre 1 e 3 e guardá-lo em R. Mande o computador para uma seqüência diferente para cada valor de R.

```
30 FOR I=1 TO 3
33 PRINT X+I*I
35 NEXT I
37 LET A=X+4*4
39 GOTO 70
40 FOR I=1 TO 3
43 PRINT I*I-Y
45 NEXT I
47 LET A=4*4-Y
49 GOTO 70
50 FOR I=1 TO 3
53 PRINT X+Y-I*I
55 NEXT I
57 LET A=X+Y-4*4
```

Para acrescentar três seqüências diferentes ao programa, apague as linhas 30-60 e acrescente um loop separado para cada seqüência. No final das primeiras duas seqüências, mande o computador para a linha 70.

Fuga do planeta Zorgo

```
 10 LET C=10
 20 PRINT "VOCE TEM ";C;" CREDITOS"
 30 PRINT "QUANTO VOCE APOSTA ";
 40 INPUT B
 50 IF B>C THEN PRINT "VOCE NAO TEM
    TANTOS CREDITOS":GOTO 20
 60 LET C=C-B
 70 LET X=INT(RND(1)*6+1)
 80 LET Y=INT(RND(1)*6+1)
 90 PRINT "APERTE P PARA JOGAR ";
100 INPUT P$
110 IF P$<>"P" THEN GOTO 90
120 PRINT TAB(5);X;TAB(10);Y
130 IF X=Y THEN GOTO 250
140 IF X+Y=10 OR X+Y=11 THEN GOTO 210
150 IF X+Y=6 OR X+Y=7 THEN GOTO 190
160 PRINT "PENA. VOCE PERDEU"
170 IF C=0 THEN GOTO 290
180 GOTO 20
190 PRINT "FICOU NA MESMA"
200 LET C=C+B:GOTO 20
210 PRINT "MUITO BEM. GANHOU O TRIPLO"
220 LET C=C+(B*3)
230 IF C>=50 THEN GOTO 320
240 GOTO 20
250 PRINT "GANHOU O DOBRO"
260 LET C=C+(B*2)
270 IF C>=50 THEN GOTO 320
280 GOTO 20
290 PRINT "SEU DINHEIRO ACABOU"
300 PRINT "NAO PODE MAIS FUGIR"
310 STOP
320 PRINT "MUITO BEM. VOCE PODE"
330 PRINT "FUGIR DE ZORGO COM ";
    C;" CREDITOS"
```

- Linha 10: C é para contar o número de créditos.
- Linha 50: Verifica se o jogador está apostando mais créditos do que tem.
- Linha 60: Calcula quantos créditos restaram.
- Linhas 70-80: Escolhe dois números aleatórios entre 1 e 6.
- Linha 110: Verifica se houve uma jogada.
- Linha 120: Mostra o resultado na tela.
- Linhas 130-150: Manda o computador para linhas diferentes, dependendo do resultado.
- Linha 170: Verifica quantos créditos restam. Se C = 0 manda o computador para o final do programa.
- Linhas 180 e 200: Manda o computador de volta para outra aposta.
- Linha 200: Atualiza o valor de C.
- Linha 230: Se C = 50 ou mais, manda o computador para o final do programa.

Jogo do Papel, Pedra e Tesoura

Modifique as linhas abaixo para que o programa funcione corretamente:

```
 50 LET R=INT(RND(1)*3+1)
 60 IF R=3 THEN LET C$="TESOURA"
150 IF C$="TESOURA" AND A$="PEDRA"
THEN LET F=1
190 PRINT "PORTANTO"
210 IF F=1 THEN PRINT "VOCE GANHOU"
230 IF F=0 THEN LET C=C+1
240 IF F=1 THEN LET A=A+1
280 IF C<10 AND A<10 THEN GOTO 40
```

Mexendo com caracteres (páginas 16-17)

Testes com strings

1

```
△40 PRINT TAB(K);MID$(N$,K,1)
```

Isto faz com que o computador imprima uma letra de cada vez, na posição K, começando com a letra número K.

2

```
 10 LET K$="CANGURU"
 20 LET L=LEN(K$)
 30 FOR J=L TO 1 STEP-1
△40 PRINT MID$(K$,J,1);
 50 NEXT J
```

Use um loop com incremento-1 para fazer o computador escrever uma palavra ao contrário.

3

```
△50 PRINT RIGHT$(S$,L-J)+LEFT$(S$,J)
```

Cada vez que o loop é repetido, RIGHT$ (S$,L-J) faz o computador imprimir as primeiras L (tamanho da palavra) menos J letras à direita de CABANAS, seguidas pelas outras J letras. Experimente rodar o programa com outras palavras.

A maior palavra

```
50 IF LEN(W$)>LEN(A$)
THEN LET A$=W$
```

Cada vez que o loop é repetido, o computador compara o tamanho de W$ com o de A$. Se W$ é maior, ele guarda W$ em A$. No final, A$ contém a maior de todas as palavras.

A menor palavra

```
10 LET A$="XXX!!!&&ABC***123!!!!
XXXXXXXXXXX"
50 IF LEN(W$)<LEN(A$)
THEN LET A$=W$
70 PRINT "MENOR PALAVRA :"
```

O programa é igual ao da palavra maior, a não ser pelas linhas 10 e 50. A linha 10 contém uma string cujo comprimento vai ser comparado com o das palavras escolhidas. Não importam os caracteres que você coloca em A$ na linha 10.

Editor de texto

Aqui estão as linhas completas:

```
40 LET S$=" "+S$+" "
70 LET W$=" "+W$+" "
```

Você precisa acrescentar espaços dos dois lados de S$ e W$ para ter certeza de que na linha 140 o computador vai procurar apenas palavras inteiras. Sem os espaços, o computador poderia escolher pedaços de palavras que fossem iguais a S$ ou W$. Experimente rodar o programa sem esses espaços para ver o que acontece.

```
100 LET LS=LEN(S$)
110 LET LW=LEN(W$)
```

As linhas acima fazem LS igual ao comprimento de S$ e LW igual ao comprimento de W$. A linha 100 é repetida na linha 160 para fazer LS igual ao comprimento da nova frase.

```
△140 IF MID$(S$,K,LW)=W$
 THEN LET A$=S$
```

Cada vez que o loop é repetido, MID$(S$,K,LW) faz o computador verificar LW (o tamanho de W$) caracteres em S$, começando no caractere K, Quando ele encontra uma seqüência de caracteres igual à que está guardada em W$, ele guarda toda a frase em uma nova variável, A$.

```
△150 IF A$=S$ THEN LET S$=LEFT$(A$,K)
 +N$+RIGHT$(A$,LS-(K+LW-2))
```

LEFT$(A$,K) separa os caracteres à esquerda da palavra que você quer substituir. A melhor maneira de compreender como esta linha funciona é experimentá-la em uma frase em um pedaço de papel.

```
180 IF K<=LS-LW+1 THEN GOTO 140
```

Mande o computador de volta à linha 140 para verificar se a palavra que você quer mudar ocorre de novo na mesma frase.

Alterações para os micros da família Sinclair

Testes com strings

Use as linhas abaixo nas respostas aos testes 1, 2 e 3.

1

```
40 PRINT TAB(K);N$(K TO K)
```

2

```
40 PRINT K$(J TO J);
```

3

```
50 PRINT S$(J+1 TO )+S$( TO J)
```

Editor de texto

```
140 IF S$(K TO K+LW-1)=W$ THEN
LET A$=S$
150 IF A$=S$ THEN LET S$=A$( TO K)
+N$+A$(K+LW-1 TO )
```

Use as linhas acima no programa Editor de texto.

Outras instruções (página 18)

Testes com letras

1 ▲
```
10 FOR K=65 TO 90
```

Estes são os números para imprimir o alfabeto no código ASCII. Para o TK-83, os números devem ser 38 a 63.

2
```
10 FOR K=97 TO 122
20 PRINT CHR$(K);
30 NEXT K
```

Estes são os números para imprimir o alfabeto em letras minúsculas. (Em alguns micros, que não têm letras minúsculas, esses números são usados para um conjunto diferente de letras maiúsculas.)

3 ▲
```
10 LET R=INT(RND(1)*26+65)
20 PRINT CHR$(R);
30 GOTO 10
```

Esta é a maneira mais simples de imprimir uma série de letras ao acaso. Para o TK-83, substitua 65, o número de código da primeira letra do alfabeto, por 38.

Como comparar letras

```
10 INPUT X$,Y$
20 IF X$<Y$ THEN PRINT X$;" VEM
ANTES DE ";Y$
30 IF Y$<X$ THEN PRINT Y$;" VEM
DEPOIS DE ";X$
40 GOTO 10
```

Se o seu micro só permite uma variável depois de INPUT, use duas instruções INPUT em linhas sucessivas.

Caixa alta/caixa baixa

```
50 IF X$>="A" AND X$<="Z" THEN PRINT
CHR$(ASC(X$)-32);
60 IF X$>="A" AND X$<="Z" THEN PRINT
CHR$(ASC(X$)+32);
```

O número que falta nas linhas 50 e 60 é 32. Esta é a diferença entre o número de código das letras maiúsculas e o das minúsculas.

Códigos secretos (página 19)

Código alfabético

Aqui está o programa completo. Se o seu micro é o TK-85, mude a instrução ASC para CODE. Se é o TK-83, use o programa à direita.

```
 10 PRINT "DIGA SUA MENSAGEM"
 20 INPUT M$
 30 FOR J=1 TO LEN(M$)
△40 LET X=ASC(MID$(M$,J,1))
 50 IF X<65 OR X>90
 THEN LET N=X:GOTO 100
 60 IF INT(J/2)=J/2 THEN LET N=X+1
 70 IF INT(J/2)<>J/2 THEN LET N=X-1
 80 IF N<65 THEN LET N=N+26
 90 IF N>90 THEN LET N=N-26
100 PRINT CHR$(N);
110 NEXT J
```

▲ Código alfabético para o TK-83

```
 10 PRINT "DIGA A SUA MENSAGEM"
 20 INPUT M$
 30 FOR J=1 TO LEN(M$)
 40 LET X=CODE(M$(J TO J))
 50 IF X<38 OR X>63 THEN LET N=X
 55 IF X<38 OR X>63 THEN GOTO 100
 60 IF INT(J/2)=J/2 THEN LET N=X+1
 70 IF INT(J/2)<>J/2 THEN LET N=X-1
 80 IF N<38 THEN LET N=N+26
 90 IF N>63 THEN LET N=N-26
100 PRINT CHR$(N);
110 NEXT J
```

As linhas 60-70 verificam se a variável J é par ou ímpar dividindo J por 2 e usando INT para transformá-lo em número inteiro. O computador então verifica se a resposta é J/2; em caso afirmativo, J é par.

Código com número-chave

Use o programa de código alfabético e acrescente linhas para escolher um número secreto. Você terá que mudar a linha 60 para somar o número-chave (K) ao número de código ASCII (X) de cada caractere, e apagar a linha 70.

```
25 PRINT "DIGA O NUMERO-CHAVE"
27 INPUT K
60 LET N=X+K
70 Apague esta linha.
```

Código variável

Este programa também é semelhante ao do código alfabético. Na linha 60, você deve somar a variável do loop (J) ao número de código ASCII de cada caractere.

```
60 LET N=X+J
70 Apague esta linha.
```

Decodificador do código variável

Para programar um decodificador, mude a linha 60 para:

```
60 LET N=X-J
```

Código de inversão

```
 10 INPUT "MENSAGEM ";M$
 20 FOR J=1 TO LEN(M$) STEP 2
△30 PRINT MID$(M$,J+1,1);
△40 PRINT MID$(M$,J,1);
 50 NEXT J
```

O incremento 2 faz o computador contar de dois em dois. Cada vez que o loop é repetido, o computador imprime o segundo (J + 1) de um par de caracteres, seguido pelo primeiro (J).

Código de inversão para os micros da família Sinclair

```
30 PRINT M$(J+1 TO J+1);
40 PRINT M$(J TO J);
```

Exercícios com INKEY$ (páginas 20-21)

Imprimindo BOM DIA

```
10 LET A$=INKEY$
20 IF A$="" THEN PRINT " ";
30 IF A$<>"" THEN PRINT "BOM DIA";
40 GOTO 10
```

Enquanto nenhuma tecla é apertada, A$ fica vazio e o computador imprime espaços (linha 20). Quando uma tecla é apertada, o caractere correspondente é guardado em A$ e o computador imprime BOM DIA (linha 30).

Use a instrução INKEY$ apropriada para o seu micro.

Fazendo o computador esperar

```
10 LET A$=INKEY$
20 IF A$="" THEN GOTO 10
30 Resto do programa...
```

A linha 20 faz o computador repetir a linha 10 enquanto nenhuma tecla for apertada.

Descubra os erros

Corrija as linhas abaixo para que o programa funcione.

Linha	Correção
`20 PRINT "APERTE QUALQUER TECLA"`	Coloque as aspas que faltam.
`50 LET X=INT(RND(1)*25+1)`	Multiplique por 25 e some 1 para ter um número aleatório entre 1 e 25.
`80 PRINT X;" + ";Y;" = "`	Tire as aspas antes de X.
`110 LET A$=INKEY$`	O nome da variável deve vir antes de INKEY$.
`140 IF N<50 THEN GOTO 90`	N deve variar de 1 a 50 para todas as respostas possíveis aparecerem na tela.
`160 FOR K=1 TO 1000:NEXT K`	Coloque a variável correta depois de NEXT.
`170 GOTO 40`	Mude o número da linha para que o valor de N volte a 0.
`190 PRINT "ACERTOU. A RESPOSTA E ";X+Y`	Os nomes das variáveis não devem ficar dentro das aspas.

Exame de motorista (página 21)

Este é o programa para o exame de motorista.

```
10 CLS
20 LET C=5
30 LET L=1
40 LET W=10
50 LET R=L+W
60 IF L<=1 THEN LET N=1
70 IF L>=25 THEN LET N=0
80 IF N=1 THEN LET L=L+1
90 IF N=0 THEN LET L=L-1
100 LET A$=INKEY$
110 IF A$=">" THEN LET C=C+1
120 IF A$="<" THEN LET C=C-1
130 PRINT TAB(L);"I";TAB(C);"*";TAB(R);"I"
140 IF C<=L OR C>=R THEN PRINT "*** BATEU ***"
150 GOTO 50
```

- Linhas 60-70: Quando L chega à extremidade da tela, o valor de N muda.
- Linhas 80-90: Quando N = 1, o valor de L aumenta de 1. Quando N = 0, o valor de L diminui de 1.
- Linhas 100-120: Estas linhas permitem que você dirija o carro.
- Linha 130: Imprime a estrada e o carro.
- Linha 140: Verifica se o carro bateu.
- Linha 150: Manda o computador de volta à linha 50 para tornar a calcular o valor de R cada vez que L varia.

Se o programa estiver muito rápido, acrescente o seguinte loop de espera:

```
142 FOR K=1 TO 400:NEXT K
```

Alterações para o TK-83

Acrescente a linha: `125 SCROLL`

Neste micro, o programa funciona melhor com o sistema de contagem. Quando você bater, deve apertar CONT para fazer o computador continuar o programa.

Sistema de contagem

```
140 IF C<=L OR C=R THEN GOTO 160
145 IF A$<>"" THEN LET S=S+1
160 PRINT "*** BATEU ***"
170 LET CR=CR+1
180 IF CR<=5 THEN GOTO 20
190 PRINT "O JOGO ACABOU"
200 PRINT "VOCE FEZ ";S;" PONTOS"
210 STOP
```

A linha 145 lhe dá 1 ponto cada vez que você dirige o carro sem bater. A linha 180 faz o programa parar depois de cinco batidas. Você precisa dar a CR (número de batidas) e S (a contagem) o valor inicial 0 no começo do programa.

Para fazer ziguezagues

```
75 IF L=5 OR L=10 THEN
LET N=INT(RND(1)*2)
76 IF L=15 OR L=20 THEN
LET N=INT(RND(1)*2)
```

Uma maneira de fazer isto é deixar de vez em quando que o computador escolha ao acaso para onde vai a estrada. Para isso, você pode mudar aleatoriamente o valor de N quando L é igual, por exemplo, a 5, 10, 15 e 20.

Quebra-cabeças com DATA (páginas 22-23)

Lista de nomes

```
50 IF X$="JORGE" THEN PRINT "SEU NOME NAO
ESTA NA LISTA":STOP
80 DATA CARLOS,LAURA,LUCIANO
90 DATA MARCOS,LUZIA,JORGE
```

Coloque os nomes que quiser em listas de DATA como as das linhas 80 e 90 acima, separando os nomes por vírgulas. Coloque o último nome da lista na linha 50. Se o seu micro não aceita linhas com mais de uma instrução, repita a instrução IF...THEN com STOP em uma outra linha (55, por exemplo).

Uso de RESTORE

```
10 FOR J=65 TO 90 ———— Códigos ASCII
20 FOR I=1 TO 19 ———— Número de dados
△40 IF LEFT$(N$,1)=CHR$(J) THEN PRINT N$
```

Alterações para a família Sinclair

```
40 IF N$(1 TO 1)=CHR$(J) THEN PRINT N$
```

Descubra os erros

1. Como os dados contêm caracteres que não são números, como - e /, você deve mudar a variável da linha 20 para uma variável string, como N$.

2. A instrução GOTO faz o computador continuar tentando ler dados mesmo depois de chegar ao final da lista de dados, de modo que vai aparecer uma mensagem de erro como "OUT OF DATA AT LINE 50" ("Dados esgotados na linha 50"). Para evitar este problema, você pode usar um loop que se repita tantas vezes quantos forem os dados ou colocar o último dado em uma instrução IF...THEN, como no programa Lista de Nomes.

Bar Bado

```
 10 PRINT "BEM-VINDO AO BAR BADO"
 20 PRINT "QUANTO PODE GASTAR"
 30 INPUT X
 40 PRINT
 50 PRINT "EIS O QUE VOCE PODE PEDIR:"
 60 PRINT
 70 READ Y$,Y
 80 IF Y<=X THEN PRINT Y$
 90 IF Y$="CAFEZINHO" THEN STOP
100 GOTO 70
110 DATA ENSOPADO DE BODE,1990.
```

Coloque todos os outros pratos e preços em linhas de DATA, deixando CAFEZINHO para último.

Catálogo telefônico

```
 10 PRINT "QUAL O NOME DA PESSOA?"
 20 PRINT
 30 INPUT N$
 40 READ X$,Y$
 50 IF N$=X$ THEN GOTO 80
 60 IF X$=" " THEN PRINT "NAO ESTA NA LISTA":GOTO 90
 70 GOTO 40
 80 PRINT "TELEFONE:";Y$
 90 PRINT
100 PRINT "QUER CONTINUAR";
110 INPUT A$
120 IF A$="SIM" THEN GOTO 150
130 IF A$="NAO" THEN STOP
140 PRINT "NAO ENTENDI":GOTO 90
150 RESTORE:GOTO 10
160 DATA ERREDOIS DEDOIS,222-3455
170 DATA PRINCESA LEIA,(033)290-8989
```

- Linha 40: Lê os nomes em X$ e os números em Y$
- Linha 50: Vai para a linha 80 quando encontra o nome.
- Linha 60: Coloque o último nome da sua lista nesta linha, entre aspas.
- Linha 140: Você precisa desta linha para o caso de alguém escrever algo que não seja sim ou não na linha 100.
- Linha 150: Mande o computador para o início da linha de DATA.
- Linha 170: Coloque os nomes e números em linhas de DATA.

Uso de matrizes (páginas 24-25)

Matrizes numéricas

```
10 DIM N(6)
50 DATA 1066,1216,1485,1603,1665,1959
```

Impressão dos dados

Você precisa usar um loop para imprimir os dados (lembre-se de mudar o número da linha de DATA):

```
50 FOR K=1 TO 6
60 PRINT "N(";K;")     VALE      ";N(K)
70 NEXT K
```

Este loop imprime ao acaso os elementos da matriz:

```
50 FOR K=1 TO 10
60 LET R=INT(RND(1)*6+1)
70 PRINT N(R)
80 NEXT K
```

Matrizes string

```
△10 DIM N$(5)
 20 FOR I=1 TO 5
▲30 READ N$(I)
 40 NEXT I
 50 FOR I=1 TO 5
 60 PRINT "N$(";I;")  TEM  ";N$(I)
 70 NEXT I
 80 DATA JORGE,CINTIA,OLGA,LEO,EMA
```

Mudanças para a família Sinclair

```
 10 DIM N$(5,7)
▲30 INPUT N$(I)
```

Calendário

Eis as linhas e variáveis que estavam faltando:

```
△10 DIM M$(12),D(12)
▲30 READ M$(K),D(K)
 70 PRINT M$(N);" TEM ";
 80 PRINT D(N);" DIAS"
```

As linhas de dados devem ser assim:

```
▲90 DATA JANEIRO,31
```

Acrescente as seguintes linhas:

```
 50 PRINT "QUAL O NOME DO MES?"
 60 INPUT A$
 62 FOR I=1 TO 12
△63 IF M$(I)=A$ THEN LET F=I
 64 NEXT I
 70 PRINT A$;" TEM ";
 80 PRINT D(F);" DIAS "
```

Mudanças para a família Sinclair

```
 10 DIM M$(12,9)
 15 DIM D(12)
▲30 INPUT M$(K)
▲35 INPUT D(K)
 63 IF M$(I)( TO LEN(A$))=A$ THEN
LET F=I
```

Programa de sugestões

```
△10 DIM I$(10)
 20 FOR K=1 TO 10
▲30 READ I$(K)
 40 NEXT K
 50 PRINT "ESCOLHA UM NUMERO DE 1 A 10"
 60 INPUT N
 70 PRINT "VA ";
 80 PRINT I$(N)
 90 PRINT "OK ";
100 INPUT A$
110 IF A$="SIM" THEN STOP
120 PRINT:GOTO 50
130 DATA PLANTAR BANANEIRA,AO CINEMA
```

Coloque o que quiser na linha 130 e seguintes.

Mudanças para a família Sinclair

```
 10 DIM I$(10, ? )
▲30 INPUT I$(K)
```

Conte o número de caracteres da sua string mais comprida e coloque o numero na instrução DIM.

Gráfico de números aleatórios

```
100 FOR K=1 TO 10
110 PRINT K;TAB(4);
120 FOR L=1 TO A(K)
130 PRINT "*";
140 NEXT L
150 PRINT
160 NEXT K
```

Uso de sub-rotinas (página 26)

Pesquisa de sorvete

```
△10 DIM I$(5),N(5)
 20 FOR K=1 TO 5
▲30 READ I$(K),N(K)
 40 GOSUB 70
 50 NEXT K
 60 STOP
 70 PRINT I$(K);TAB(11);" ";
 80 FOR L=1 TO  N(K)
 90 PRINT "*";
100 NEXT L
110 PRINT
120 RETURN
130 DATA MANGA,1,ABACAXI,11
140 DATA COCO,8,MORANGO,1
150 DATA GRAVIOLA,18
```

Você precisa da instrução TAB na linha 70 para que o computador imprima asteriscos no mesmo lugar em todas as linhas.

Mudanças para a família Sinclair

```
 10 DIM I$(5,10)
 15 DIM N(5)
▲30 INPUT I$(K)
▲35 INPUT N(K)
```

Afunde o submarino

```
200 PRINT "ERROU"
210 PRINT "VA PARA ";
220 IF B=Y THEN GOTO 250
230 IF B<Y THEN PRINT "N ";
240 IF B>Y THEN PRINT "S ";
250 IF A=X THEN GOTO 280
260 IF A<X THEN PRINT "E"
270 IF A>X THEN PRINT "O"
280 RETURN
```

Esta é a sub-rotina para comparar a posição do submarino com o palpite e imprimir uma mensagem na tela.

Vinte perguntas

```
△10 DIM Q$(20),A(20)
 20 FOR K=1 TO 20
▲30 READ Q$(K),A(K)
 40 NEXT K
 50 FOR L=1 TO 20
 60 PRINT Q$(L)
 70 INPUT X
 80 IF X=A(L) THEN PRINT "CERTO"
 90 IF X<>A(L) THEN PRINT "IGNORANTE!
 A RESPOSTA E ";A(L)
100 PRINT
110 NEXT L
120 DATA QUANTAS PERNAS TEM UMA CENTOPEIA?,100
130 DATA QUANTOS JOGADORES TEM
UM TIME DE VOLEI?,5
```

Mudanças para a família Sinclair

```
 10 DIM Q$(20,?)
 15 DIM A(20)
▲30 INPUT Q$(K)
▲35 INPUT A(K)
```

Conte o número de caracteres da sua string mais comprida e coloque esse número na instrução DIM.

Programa da máquina caça-níqueis

```
10 DIM F$(6)
20 FOR K=1 TO 6
30 READ F$(K)
40 NEXT K
50 CLS
60 LET N=10
70 PRINT "VOCE TEM ";N;" MOEDAS"
80 LET I$=INKEY$
85 PRINT "APERTE QUALQUER TECLA:";
90 IF I$="" THEN GOTO 80
100 LET N=N-1
110 CLS
120 LET A=INT(RND(1)*6+1) ┐
130 LET A$=F$(A)          │
140 LET A=INT(RND(1)*6+1) ├ Escolhe três frutas em F$.
150 LET B$=F$(A)          │
160 LET A=INT(RND(1)*6+1) │
170 LET C$=F$(A)          ┘
180 PRINT
190 PRINT A$;"  ";B$;"  ";C$ ]—— Imprime as frutas.
200 PRINT
210 IF A$=B$ AND B$=C$ AND            ┐
    C$="CEREJAS" THEN GOSUB 270       │
220 IF A$=B$ AND B$=C$ AND            │
    C$<>"CEREJAS" THEN GOSUB 310      ├── Verifica qual
230 IF (A$=B$ AND B$<>C$) OR          │   foi o prêmio.
    (A$=C$ AND C$<>B$) OR (B$=C$      │
    AND C$<>A$) THEN GOSUB 350        ┘
240 IF N>0 THEN GOTO 70
250 PRINT "SEU DINHEIRO ACABOU"
260 STOP
270 PRINT "3 CEREJAS"                  ┐
280 PRINT "VOCE GANHOU 20 MOEDAS"      ├── 1ª sub-rotina.
290 LET N=N+20                         │
300 RETURN                             ┘
310 PRINT "3 IGUAIS"                   ┐
320 PRINT "VOCE GANHOU 5 MOEDAS"       ├── 2ª sub-rotina.
330 LET N=N+5                          │
340 RETURN                             ┘
350 PRINT "2 IGUAIS"                   ┐
360 PRINT "VOCE GANHOU 2 MOEDAS"       ├── 3ª sub-rotina.
370 LET N=N+2                          │
380 RETURN                             ┘
390 DATA LIMAO,CEREJAS,MELANCIA
400 DATA SINO,UVAS,AMEIXA
```

Mudanças para a família Sinclair

```
 10 DIM F$(6,6)
▲20 INPUT F$(K)
```

Programa de caça ao tesouro (páginas 28-33)

Aqui está o programa completo do jogo de caça ao tesouro, listado parte por parte, na ordem correta para ser executado. As alterações para a família Sinclair estão no final.

1 Definição das matrizes e leitura dos dados

```
△ 100 DIM N(7),L(7),S(7),O(7),D$(7),
T$(7),T(7)
110 FOR K=1 TO 7
▲ 120 READ N(K),L(K),S(K),W(K)
130 NEXT K
140 FOR K=1 TO 7
▲ 150 READ D$(K)
160 NEXT K
170 FOR K=1 TO 7
▲ 180 READ T$(K),T(K)
190 NEXT K
200 LET M=0
210 LET C=0
220 LET F=0
230 LET W=0
240 LET X=0
250 LET Y=0
260 CLS
```

Os dados estão da linha 2000 em diante.

2 Sub-rotina Ajuda

```
300 GOSUB 1000
```

A sub-rotina está da linha 1000 em diante.

3 Escolha de um quarto ao acaso

```
△ 350 LET R=INT(RND(1)*7+1)
```

4 Identificação do quarto

```
400 PRINT "VOCE ESTA NO QUARTO ";R
410 PRINT "ELE E ";D$(R)
420 PRINT "ELE CONTEM ";
```

Identificação do conteúdo

```
430 FOR K=1 TO 7
▲ 440 IF T(K)=R THEN PRINT TAB(15);
T$(K):LET F=1
450 NEXT K
460 IF F=0 THEN PRINT TAB(15)"NADA"
470 LET F=0
▲ 480 PRINT:PRINT
```

F é uma variável "bandeira". Na linha 440, a bandeira "sobe" (F = 1) quando um dos números da matriz T é igual ao número do quarto (R).

5 Jogada

```
500 PRINT "DIGA O QUE QUER"
510 INPUT A$
520 IF A$="AJUDA" THEN GOSUB 1000
530 IF A$="N" OR A$="L" OR A$="S" OR
A$="O" THEN GOSUB 1200
540 IF A$="PEGAR" THEN GOSUB 1300
550 IF A$="LARGAR" THEN GOSUB 1400
560 IF A$="LOCAL" THEN GOSUB 1500
```

As linhas 520 a 560 mandam o computador para diferentes sub-rotinas, dependendo da palavra que o jogador coloca na variável A$ (linha 510).

10 Contagem das jogadas

```
600 LET M=M+1
```

11 Todas as caixas estão no mesmo quarto?

```
610 LET W=T(1)
620 FOR K=2 TO 7
630 IF W<>T(K) THEN LET F=1
640 NEXT K
650 IF F=1 THEN GOTO 690
660 PRINT "MUITO BEM. VOCE LEVOU
TODO O TESOURO"
670 PRINT "PARA O QUARTO ";R;
" EM ";M;" JOGADAS"
680 STOP
690 LET F=0
```

Na linha 610, o número que está em T(1) é guardado na variável W. Dentro do loop (linha 630) o computador compara W com todos os outros números que estão guardados na matriz T. Se qualquer dos elementos de T é diferente de W, o computador faz a variável bandeira (F) igual a 1. Se todos os elementos de T são iguais a W, F continua a valer 0, o que significa que todas as caixas estão no mesmo quarto.

12 O jogador usou todas as suas jogadas?

```
700 IF M<=28 THEN GOTO 730
710 PRINT "PENA. SUAS JOGADAS ACABARAM."
720 STOP
730 PRINT
```

Se M é maior que 29, as jogadas acabaram e o programa pára. Você pode mudar à vontade o número de jogadas permitido.

13 O jogador mudou de quarto?

```
800 IF A$="N" OR A$="L" OR A$="S" OR
A$="O" THEN GOTO 400
810 PRINT "VOCE AINDA ESTA NO QUARTO ";R
850 GOTO 500
```

2 Sub-rotina Ajuda

```
1000 PRINT "EXISTEM SETE QUARTOS NO
LABIRINTO"
1010 PRINT "E EXISTE UMA CAIXA EM"
1020 PRINT "CADA QUARTO. VOCE TEM QUE
LEVAR"
1030 PRINT "TODAS AS CAIXAS PARA O MESMO
QUARTO"
1040 PRINT
1050 PRINT "ESTAS SAO AS PALAVRAS QUE O
COMPUTADOR COMPREENDE"
1055 PRINT
1060 PRINT "N,S,L,O : ANDAR PARA O NORTE,SUL,
LESTE OU OESTE"
1070 PRINT "PEGAR   : PEGAR CAIXA"
1080 PRINT "LARGAR  : LARGAR CAIXA"
1090 PRINT "LOCAL   : MOSTRA A LOCALIZACAO
DAS CAIXAS"
1100 PRINT "AJUDA   : MOSTRA AS REGRAS DO JOGO"
▲ 1110 PRINT:PRINT
1120 RETURN
```

O computador diz a você quais são as regras do jogo e quais as palavras que ele compreende. Cada vez que o jogador escreve a palavra AJUDA o computador mostra essas informações.

6 Sub-rotina Andar

```
1200 IF A$="N" THEN LET X N(R)
1210 IF A$="L" THEN LET X E(R)
1220 IF A$="S" THEN LET X S(R)
1230 IF A$="O" THEN LET X W(R)
▲1240 IF X=0 THEN PRINT "MOVIMENTO
IMPOSSIVEL":GOTO 1260
1250 LET R=X
1260 RETURN
```

Nas linhas 1200-1230, o computador verifica em que direção o jogador deseja ir (N,L,S, ou O). Em seguida, descobre qual é o quarto nessa direção, usando R como índice da matriz apropriada, e guarda a informação em X. Se X é 0 (linha 1240), isso quer dizer que não há nenhum quarto na direção considerada.

7 Sub-rotina Pegar

```
▲1300 IF C=1 THEN PRINT "NAO PODE
CARREGAR MAIS NADA:GOTO 1370
1310 FOR K=1 TO 7
1320 IF T(K)=R THEN LET Y=K
1330 NEXT K
1340 IF Y=0 THEN PRINT "ESTE QUARTO
ESTA VAZIO"
1350 LET T(Y)=999
1360 PRINT "OK. VOCE ESTA CARREGANDO ";T$(Y)
▲1370 LET C=1:LET Y=0
1380 RETURN
```

Na linha 1320, o computador compara todos os números guardados na matriz T com R. Se um deles é igual, toma o número do índice do elemento de T correspondente e o guarda na variável Y. Depois do loop, o computador usa Y para escolher o elemento apropriado de T e muda o seu valor para 999 (linha 1350). Se Y = 0 (linha 1340), isto significa que o quarto está vazio.

8 Sub-rotina Largar

```
▲1400 IF C=0 THEN PRINT "VOCE NAO
ESTA CARREGANDO NADA":GOTO 1450
1410 FOR K=1 TO 7
1420 IF T(K)=999 THEN PRINT T$(K);
" : DEIXOU NO QUARTO ";R
1430 IF T(K)=999 THEN LET T(K)=R
1440 NEXT K
1450 LET C=0
1460 RETURN
```

O computador verifica qual o elemento de T que e igual a 999 (linha 1420). Em seguida, muda seu valor para R (linha 1430).

9 Sub-rotina Local

```
1500 PRINT "VOCE ESTA CARREGANDO ";
▲1510 IF C=0 THEN PRINT TAB(10);
"NADA":GOTO 1550
1520 FOR K=1 TO 7
1530 IF T(K)=999 THEN PRINT TAB(10);T$(K)
1540 NEXT K
1550 PRINT "CONTEUDO DOS QUARTOS: "
1560 FOR K=1 TO 7
1570 IF T(K)<>999 THEN PRINT T(K);" : ";T$(K)
1580 NEXT K
1590 RETURN
```

1 Dados

```
▲2000 DATA 2,7,6,0
2010 DATA 0,3,7,1
2020 DATA 0,0,4,2
2030 DATA 3,0,5,7
2040 DATA 7,4,0,6
2050 DATA 1,5,0,0
2060 DATA 2,4,5,1
```

Estes são os dados para as matrizes N,S,L e O.

```
2100 DATA FRIO E FEDORENTO,PRETO
E PESTILENTO
2110 DATA TORTO E TENEBROSO,
SOMBRIO E SEPULCRAL
2120 DATA VAZIO E VENENOSO,AMPLO
E ASSUSTADOR,GRANDE E GOSMENTO
```

Estes são os dados para D$

```
2200 DATA OURO,1,CHICLETES,2
2210 DATA SANDUICHES,3,LIXO,4
2220 DATA POTES DE MEL,5,JOIAS,6,
MOEDAS,7
```

Estes são os dados para T$ e T.

Alterações para a família Sinclair

A única alteração para o TK-85 é que as matrizes string devem ser dimensionadas como para o TK-83. Para o TK-83, devem ser feitas as seguintes modificações:

Linha 100: Coloque as instruções DIM em linhas separadas e dimensione as matrizes string como D$(7,20) e T$ (7,23).

Linha 120: Use linhas separadas de INPUT para cada matriz e entre com os dados quando rodar o programa.

```
120 INPUT N(K)
123 INPUT L(K)
125 INPUT S(K)
127 INPUT O(K)
```

Linha 150: Use INPUT e entre com os dados quando rodar o programa.

```
150 INPUT D$(K)
180 INPUT T$(K)
```

Linha 180: Use INPUT para a matriz T$ e LET para a matriz T para que você não tenha que entrar com as localizações iniciais das caixas cada vez que rodar o programa. Use um novo loop nas linhas 193-197 para a instrução LET, assim:

```
193 FOR K=1 TO 7
195 LET T(K)=K
197 NEXT K
```

Linha 350: Use a função RND correta para o seu micro:

```
350 LET R=INT(RND*7+1)
```

Linha 440: Use linhas separadas e repita a instrução IF...THEN, assim:

```
440 IF T(K)=R THEN PRINT TAB(15);T$(K)
445 IF T(K)=R THEN LET F=1
```

Linhas 480 e 1110: Use linhas separadas.

Linhas 1240-1510: Use linhas separadas e quando houver uma instrução IF...THEN, repita-a na nova linha, assim:

```
1240 IF X=0 THEN PRINT "MOVIMENTO
IMPOSSIVEL"
1245 IF X=0 THEN GOTO 1260
```

Linha 2000: Remova as linhas 2000 a 2220 e acrescente as linhas abaixo, para não ter que entrar novamente com os dados cada vez que for rodar o programa. Quando quiser jogar de novo, entre com GOTO 2000. (Se você apertar RUN, o computador apagará todas as variáveis e matrizes.)

```
2000 PRINT "QUER JOGAR DE NOVO"
2010 INPUT R$
2020 IF R$="SIM" THEN GOTO 193
2030 IF R$="NAO" THEN STOP
2040 GOTO 2000
```

Instruções de BASIC

Aqui está uma lista das instruções de BASIC usadas neste livro, com uma rápida explicação a respeito do seu significado. As instruções marcadas com um asterisco variam de acordo com a marca e o modelo do micro. A tabela de equivalência da página 47 o ajudará a descobrir a instrução correta para o seu computador.

ABS diz ao computador para ignorar o sinal de mais ou de menos na frente de um número. Assim, por exemplo, ABS(-5) é igual a 5.

***ASC** fornece o número de código ASCII de um caractere. Assim, por exemplo, ASC("A") é igual a 65, que é o número de código ASCII para a letra A. Os micros da família Sinclair usam CODE em lugar de ASC.

**CHR$** transforma um número em um caractere, de acordo com os números de código que representam os caracteres no interior do computador. Nos computadores que usam o código ASCII, CHR$(65) fornece a letra A. No TK-83, que usa um código diferente, A é dado por CHR$(38).

CLS apaga a tela.

***CODE** é usado nos micros da família Sinclair em lugar de ASC, para obter o número de código de um caractere. O TK-85 usa o código ASCII, mas o TK-83 tem um código próprio.

***DATA** vide READ/DATA

***DIM** diz ao computador para reservar um certo espaço na memória para uma matriz. Para isso, a palavra DIM é seguida pelo nome da matriz e o número de elementos que contém. Assim, por exemplo, DIM A(5) significa que a matriz A tem cinco elementos. No caso de matrizes string em micros da família Sinclair, deve ser indicado também o número de caracteres da string mais comprida.

FOR/TO...NEXT faz o computador repetir as instruções entre as linhas FOR/TO e NEXT um certo número de vezes. Isto é chamado de loop.

GOSUB diz ao computador para sair do programa principal e passar para uma parte do programa chamada de sub-rotina. GOSUB deve ser seguida pelo número da primeira linha da sub-rotina. No final da sub-rotina, a instrução RETURN diz ao computador para voltar ao programa principal, a partir da instrução seguinte a GOSUB.

GOTO faz o computador saltar para a linha cujo número se segue à palavra GOTO.

IF...THEN diz ao computador para fazer uma comparação e agir de acordo com o resultado. Depois da palavra IF, existe uma condição que o computador deve testar comparando informações. Se a condição é verdadeira, o computador executa as instruções que se seguem à palavra THEN; se é falsa, passa para a linha seguinte.

***INKEY$** verifica o teclado para ver se alguma tecla está sendo apertada. Não espera que o operador aperte alguma tecla, como a instrução INPUT, nem necessita de que a tecla RETURN (ENTER ou NEWLINE, dependendo do micro) seja apertada. A instrução muitas vezes é colocada em um loop de espera, para que o operador tenha tempo de agir.

INPUT diz ao computador para esperar que o operador introduza dados através do teclado durante a execução do programa. A palavra INPUT deve ser seguida pelo nome de uma variável.

INT é a abreviação de inteiro. A instrução transforma um número real (número com algarismos depois do ponto decimal) em um número inteiro, desprezando tudo que se segue ao ponto decimal. Assim, por exemplo, INT(6.732) = 6. No caso de números negativos, tudo que está à direita do ponto é ignorado e o número que está à esquerda é "aumentado" de uma unidade. Assim, por exemplo, INT(-3.2) = -4.

***LEFT$** diz ao computador para tomar um certo número de caracteres a partir do lado esquerdo de uma string. Assim, por exemplo, LEFT$(A$,4) diz ao computador para tomar os primeiros quatro caracteres de A$. Os micros da família Sinclair não usam esta instrução.

LEN fornece o comprimento de uma string, ou seja, o número de caracteres da string, incluindo espaços e sinais de pontuação.

LET é usado para dar um nome a um espaço na memória e guardar alguma informação nesse espaço.

LIST diz ao computador para mostrar na tela uma listagem do programa.

***MID$** diz ao computador para tomar um certo número de caracteres do meio de uma string. Assim, por exemplo, MID$(A$,5,2) diz ao computador para tomar dois caracteres da variável A$, começando pelo quinto. Os micros da família Sinclair não usam esta instrução.

NEXT diz ao computador para voltar ao começo de um loop. Vide FOR/TO...NEXT.

NEW diz ao computador para apagar o programa.

PRINT diz ao computador para mostrar alguma coisa na tela. A instrução PRINT sozinha faz o computador pular uma linha.

***READ/DATA** diz ao computador para tomar as informações contidas em linhas iniciadas com a palavra DATA e guardá-las nas variáveis ou na matriz indicada na instrução READ.

REM é uma instrução usada para colocar comentários no meio de um programa. As linhas que começam por REM são ignoradas durante a execução do programa.

RESTORE diz ao computador para voltar ao começo das linhas de DATA.

RETURN é a instrução usada no final das sub-rotinas para dizer ao computador para voltar ao programa principal.

***RIGHT$** diz ao computador para tomar um certo número de caracteres a partir do lado direito de uma string. Assim, por exemplo, RIGHT$(A$,4) diz ao computador para tomar os últimos quatro caracteres de A$. Os micros da família Sinclair não usam esta instrução.

***RND** diz ao computador para escolher um número ao acaso.

RUN diz ao computador para começar a executar um programa.

***SPC** diz ao computador para pular um certo número de espaços na tela. Nem todos os micros usam esta instrução.

STEP é usado nos loops com FOR/TO...NEXT para indicar ao computador qual o valor do incremento da variável de contagem do loop.

***TAB** faz o cursor se deslocar um certo número de espaços na tela. Geralmente é usada junto com PRINT, para mostrar alguma coisa no meio da tela. Em alguns micros, TAB também pode ser usada para fazer o cursor se deslocar verticalmente. Na maioria dos micros, a instrução TAB deve ser seguida de ponto-e-vírgula.

THEN é usada com IF para dizer ao computador o que fazer se certas condições forem satisfeitas. Vide IF...THEN.

Símbolos de BASIC

*	Multiplicação
/	Divisão
SQR	Raiz quadrada
^	Exponenciação. Este símbolo varia de acordo com a marca e o modelo de micro.
>	Maior que
<	Menor que
>=	Maior que ou igual a
<=	Menor que ou igual a
<>	Diferente de

Tabelas de conversão

Instrução usada neste livro	CLS Apaga a tela.	INT(RND(1)*N+1) Escolhe um número aleatório inteiro entre 1 e N.	INKEY$ Faz o computador verificar se alguma tecla foi apertada.
TRS 80	CLS	INT(RND(0)*N+1)	INKEY$
APPLE	HOME	INT(RND(1)*N+1)	X$="" IF PEEK(-16384) >127 THEN GET X$
TK-83	CLS	INT(RND*N+1)	INKEY$
TK-85	CLS	INT(RND*N+1)	INKEY$

Conversões adicionais para os micros da família Sinclair

LEFT$ RIGHT$ MID$	Os micros na família Sinclair não processam strings da mesma forma que os outros micros. Para converter essas instruções, você precisa primeiro determinar a posição do primeiro e do último caractere que deseja tomar. Coloque os números entre parênteses, separados pela palavra TO, depois do nome da variável string correspondente, conforme os exemplos à direita.	LET X$="COMPUTADOR" PRINT X$(1 TO 4) COMP PRINT X$(5 TO) UTADOR PRINT X$(3 TO 5) MPU
ASC	Use CODE em lugar de ASC. O TK-83 tem um código próprio, mas o TK-85 usa o código ASCII.	PRINT CODE("A")
DIM	Para dimensionar uma matriz string, use DIM seguido por dois números entre parênteses. O primeiro é o número de elementos (strings) da matriz; o segundo, o número de caracteres da string mais comprida. No exemplo à direita, a matriz A$ contém cinco strings, sendo que a mais comprida tem 12 caracteres.	DIM A$(5.12)
READ/DATA	O TK-85 usa READ/DATA como os outros micros, mas o TK-83 não possui nenhuma instrução equivalente. A melhor alternativa é usar várias instruções INPUT, como no exemplo à direita.	10 FOR K=1 TO 5 20 INPUT A$(K) 30 NEXT K

Indice Remissivo

Neste índice você encontrará os nomes dos problemas, das instruções de BASIC e das idéias discutidas neste livro. Os números das páginas das respostas dos problemas estão impressos em negrito.

Título original inglês
PRACTISE YOUR BASIC

Copyright © 1983 by
Usborne Publishing Ltd.
Direitos de publicação exclusiva
em língua portuguesa
em todo o mundo
adquiridos pela
EDITORA LUTÉCIA LTDA.
Rua Argentina 171 —
20921 Rio de Janeiro, RJ —
Tel.: 580-3668
que se reserva a propriedade
literária desta tradução

Impresso no Brasil

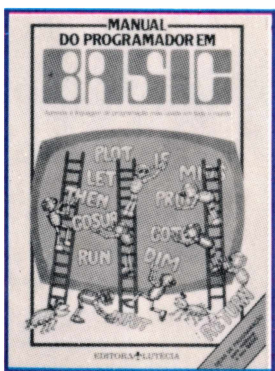
PEDIDOS PELO REEMBOLSO POSTAL
Caixa Postal 23.052 — Rio de Janeiro, RJ 20922

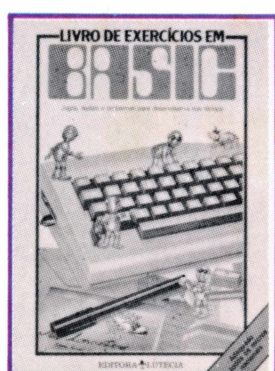
Guias Práticos de Microcomputadores

O que é capaz de fazer um microcomputador! Calcular, evidentemente, mas também organizar perguntas, escrever poemas, jogar uma quantidade de jogos cada qual mais palpitante, até mesmo compor música... Pequenos guias práticos de introdução à microinformática, as obras desta nova coleção nos permitem descobrir todas as possibilidades que os microcomputadores nos oferecem. Eles nos iniciam na linguagem e no funcionamento do computador, na aprendizagem da programação e — por que não? — na criação de programas originais: a clareza dos textos, a alegria das cores, a graça dos desenhos, tudo é concebido nestes livros para fazer desta iniciação um prazer.

Um colorido guia para melhor compreendermos os microcomputadores — como trabalham e o que fazem, apresentando idéias de coisas que podem ser feitas com o micro. Apresenta um utilíssimo guia comparativo dos diversos micros existentes no mercado brasileiro.

Um guia passo a passo de programação para os absolutamente iniciantes. Apresenta as diferentes linguagens e aprofunda no BASIC, incluindo pequenos programas, simples e divertidos. Profusamente ilustrado a cores.

13 programas fascinantes e ao mesmo tempo simples, para todo tipo de microcomputadores; Jogos Intergaláticos, Módulo Lunar, Viagem ao Futuro, Salvamento no Espaço e muitos outros, com respostas dos problemas.

Introduz o leitor na programação BASIC, através de exemplos didáticos, e é sobretudo indicado ao jovem que começa a programar. Substitui com vantagens os manuais dos fabricantes. Apresenta dezenas de dicas utilíssimas para aprimorar a técnica de programação.

13 listagens de programas de jogos, para serem "rodados" nos micros pessoais mais difundidos no Brasil. Muito útil para ensinar o leitor a desenvolver seus próprios programas. Além disso, é um passatempo fascinante.

Um verdadeiro caderno a cores de problemas e exercícios, com jogos e quebra-cabeças, ideais para todo curso de programação em BASIC. Todas as respostas e explicações estão incluídas, para ajudar a esclarecer todo e qualquer problema.